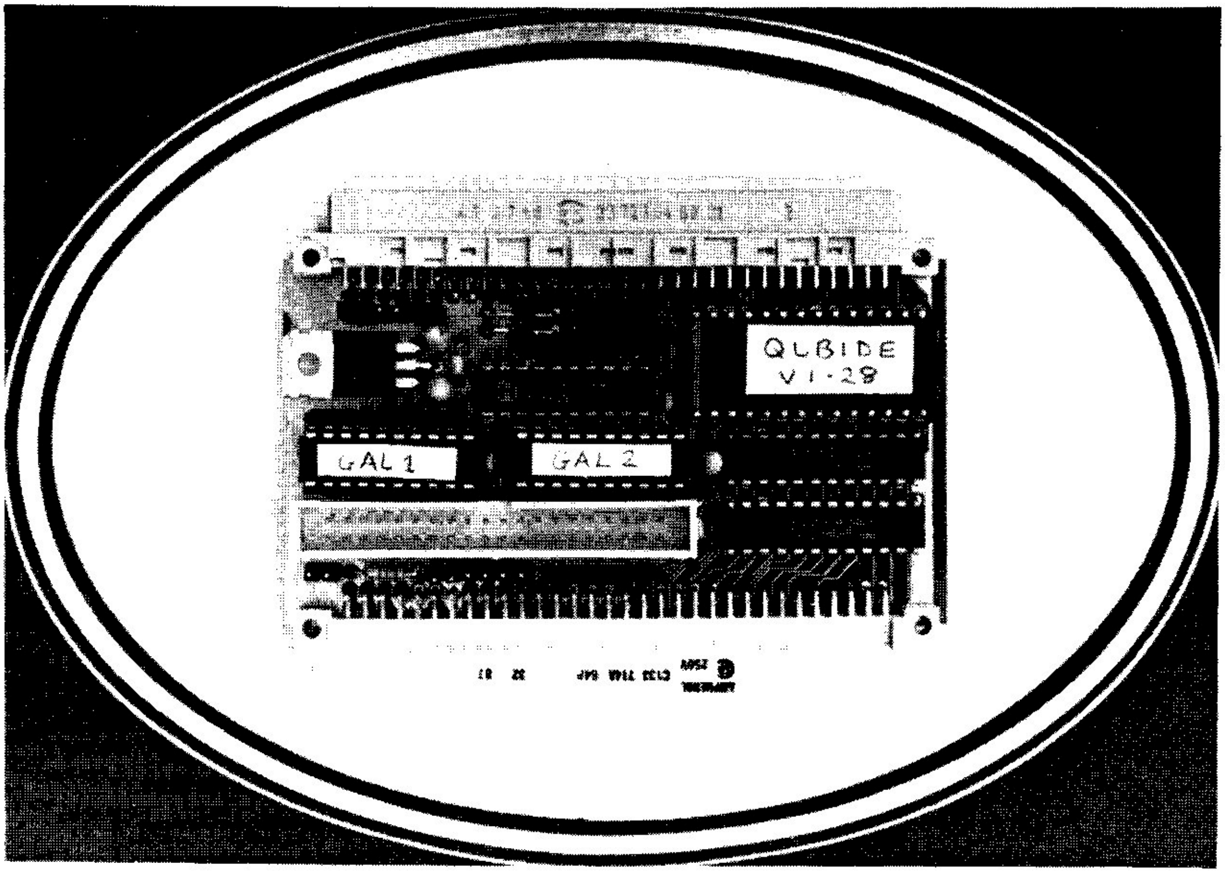
IQLR

International QL Report

ISSN 1078-5787

Volume 4 Issue 4
November/December
1994

The QL's IDE Hard Disk Interface Has Arrived !



QUBIDE from QUBBESoft P/D

IQLR......

International QL Report (ISSN 1078-5787) is published by:

IQLR P. O. Box 3991 Newport, RI 02840-0987 USA	IQLR 23 Ben Culey Drive Thetford, Norfolk IP24 1QJ GREAT BRITAIN
PUBLISHER: Robert Dyl, Sr.	Tel/Fax: +1 401 849 3805

IQLR is published bi-monthly, our volume year begins on 1 May and runs through 30 April. Subscriptions begin with the current issue at the time of sign up. Subscription rates are as follows:

USA	$20.00 per year
British Isles & Europe	£25.00 per year
Canada	$23.00 (US Funds)
Central/South America	$34.00 (US Funds)
Rest of World	$40.00 (US Funds)

UK and European readers may send their subscriptions to our European office listed above. Postal, Euro, Bank and Personal Cheques in Pounds Sterling, drawn on a UK bank should be made payable to IQLR.

Payment in US $ can be made by either a Postal, Bank or Personal cheque (drawn on a US BANK) or bank notes (£ or DM equivalent to the US $ amount) should be sent to our North American office.

We welcome your comments, suggestions and articles. YOU make IQLR possible. We are constantly changing and adjusting to meet your needs and requirements. Articles submitted for publication should be on a 3.5" disk in Quill or Text87 format. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hard copy of all screens to be included, don't forget to specify where in the text you would like the screens placed.

Article and Advertising DEADLINES are as follows:

Issue 1	10 April
Issue 2	10 June
Issue 3	10 August
Issue 4	10 October
Issue 5	10 December
Issue 6	10 February

IQLR reserves the right to publish or not publish any material submitted. Under no circumstances will IQLR be held liable for any indirect, direct or consequential damage or loss arising out of the use and/or inability to use any of the material published in IQLR. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

IQLR is produced using the following equipment: a QL computer with a Super Gold Card , Masterpiece Enhanced Graphics Card, QUBide IDE Interface, Minerva MK II ROM, Hermes Co-processor, QIMI Mouse Interface and Keyboard-90 Interface. Masters are produced on a Hewlett Packard DeskJet 520 printer. Text is generated using Textxt87 Plus4.

Contents...........

Advertisers......

EDITORIAL

Newport, Rhode Island, USA - The Editorial Staff

Changes ! Changes ! Changes!

In our attempt to continually improve our magazine, we pay a lot of attention to what our readers have to say. Many of you stated that the print style we used was too big, while others stated that we should use more of the page as we did in our last issue. Some of you wanted more space between lines so text would be easier to follow. Many asked for more in the way of graphics and screens. But, most important, is the fact that our readers take an active part in determining the esthetic, as well as the material content of IQLR. We thank you for all your help and encouragement.

As you can see, we've incorporated your ideas into this issue, let us know what you think, your responses will determine if we keep this format or return to our old format. Without prejudicing your choice, there are advantages to this new format, namely, we can pack more material in fewer pages (saving a few trees per issue).

On another front, we have received a number of inquires concerning Digital Precision and long standing orders for goods. Please note elsewhere in this issue the article "NEWS FROM DIGITAL PRECISION", it should answer most of the questions raised. The one point that is not clear is, without telephone support for DP's products, will support be available by mail (Freddy please enlighten us).

Software Library

As you know, we recently launched our library of Freeware, Shareware and Public Domain software. Included with this issue, you should have received a printed catalogue of the library offerings. We decided not to bind or punch holes in it to allow you to keep it in a manner suitable to you. In the future, we will not use valuable space in the magazine to list the new additions, instead, we will include a page(s) of additions with your issue of IQLR.

Our library has grown to its present size through the generosity of Steve Johnson of S.J.P.D. Software. Thanks, Steve! We would also like to thank the many authors who freely offer their software on the QBOX network of bulletin boards. Thank you all!

We are always looking for good quality software to add to our library, why not send us that program you developed for yourself, we'll help you share it with the entire QL community.

In this issue is an overview of Bill Cable's QLERK (financial package) by John Taylor of Quanta fame, read it, then get a demo version from our library or from the QBOX network. (Commercial suppliers - are your demo's in our library?)

Issue Delivery

The US Postal Service has failed us again, their apologies did not make a difference in their service. Our last issue was delayed AGAIN in the UK and some areas of Europe because they sent our packages by Sea when we paid for Airmail. Well, it WILL NOT HAPPEN AGAIN, as they are not getting another chance. We have contracted with UPS (United Parcel Service) for delivery to the UK and Europe, with their written guarantee of a five working day delivery schedule. With this new service, we are optimistic that all subscribers will receive their issues within a few days of one another rather than weeks. As in everything we do, we'll do our best, to improve in this area too.

Subscriptions

Our subscription base continues to grow (200 new subscribers since QL Worlds demise) and we'd like to take this opportunity to welcome our new readers to the IQLR family.

In this issue we have included an IQLR subscription flyer. Its purpose is twofold, first it contains some background information on how all this started and second, we hope you will give it to a QL friend who is not yet an IQLR subscriber (there are still thousands of QL users who don't know about IQLR). We thank you for your help.

SMSQ An EXPLANATION

Duisburg, GERMANY - Jochen Merz

SMS2 - SMSQ - SMSQ/E

There seems to be a lot of confusion about the three systems listed above, understandable perhaps, as the initials on their own mean very little to the uninitiated. I will try to explain the differences without going into too much technical detail. If it is still not clear, let me know and I'll try again!

The initials SMS stand for Small Microcomputer System. This is an operating system which can replace QDOS, with three variants for use in different operating environments.

SMS2

This was the first variant to be marketed and is a simplified, cut-down replacement for QDOS on low-end Atari ST series computers, and comes in a plug-in-and-go form. For the Atari user who doesn't know QDOS, it offers new facilities: ease of use, fast multitasking and network via Midinet. New "Atari Only" customers are generally very satisfied because they had never had the full range of facilities offered by the QL and its software extensions.

SMS2 was shown to QL users and to those who owned a QL-Emulator for the Atari ST but , for these users there are disadvantages as the system setup is not as flexible as they might expect it to be: parts of it, e.g. QPAC2 cannot be configured and most important, there is no Basic Interpreter. The majority of those who owned SMS2 and a QL-Emulator have gone back to using QDOS on the emulator, because it offered more of the facilities they were used to and which were important for running their software.

SMSQ

SMSQ for the QXL intends to be compatible with QDOS but at the same time offers features not provided by QDOS. As a replacement operating system, it runs faster and has an improved SuperBasic interpreter. Its performance and capabilities are limited by its strick compatibility requirements and for this reason it is not possible to include an integrated Pointer Interface or the more advanced drivers that have been developed for the Atari ST.

SMSQ/E

SMSQ/E is an amalgamation of SMSQ, the Atari LEVEL E drivers available for the Atari QL Emulator card(s) and the new DV3 DISK DRIVERS. This then aims to provide the best of all worlds to as many users as possible, resulting in a uniform system across all hardware platforms. Users and programmers alike should have their life made easier while allowing them to run the same software on various machines and a variety of systems: ATARI, QL with GOLD CARD, a QL with SUPER GOLD CARD or a PC with a QXL card.

It would be difficult to discuss all the advantages in this article . As an alternative, you can read about them in my adverts and in a follow up article to this one where I'll explain what all the major and minor improvements will allow you to do.

What does this all mean to you?

QL USER- sorry, but the basic QL hardware is not suitable for any current variant of SMS.

QL USER with GOLD CARD or SUPER GOLD CARD - you can extend your system with SMSQ/E and benefit from the additional facilities available now and others under development.

ATARI QL EMULATOR USER - you can upgrade your system to SMSQ/E and have the advantages of the new operating system, the multiple and extremely fast SBASIC, Level Three Device Driver and support of the ATARI TT.

SMSQ An EXPLANATION - (CONT'D)

CONCLUSION

Once you have used SMSQ/E I'm convinced that you'll never want to go back to QDOS . You'll just not want to be without all the advantages you've gained. SMSQ/E incorporates a lot of facilities users have been requesting over the years and future development promises to be exciting.

Arcwind/QL World

Months after IQLR ended its negotiations with Arcwind, we are still receiving inquires concerning unfulfilled subscriptions to QL World. We DID NOT assume any of the assets or liabilities of QL World.

If you believe, as a QL World subscriber, that you have a claim concerning your undelivered subscription, we suggest that you write to Arcwind at the address listed below:

Arcwind Ltd
The Blue Barn, Tew Lane
Wootton, Woodstock
Oxon. OX7 1HA
Great Britain

Our only contacts were with the publisher Mark Kasprowicz and not having been informed as to the present status of QL World, we are unable to list a contact person at Arcwind.

XCHANGE (IN THE PUBLIC DOMAIN AND WISH LISTS)

Birmingham, GREAT BRITAIN - Mike Bedford-White

Following on from the rather successful launch of XCHANGE onto the public domain by this group our own Simon Goodwin has had some feedback concerning problems and modifications, he has shown me some additions and alterations which have made some improvements! I won't at the moment go into what the changes are, because I want to ask the readers and possesors of XCHANGE to send me their thoughts and wish lists in the expectation that some or all of the suggestions could be incorporated into XCHANGE.

I know that there is a lot of interest because of the response I have had for copies of XCHANGE from me and the letters that have appeared in QUANTA magazine and IQLR.

Along with the wish lists, I would also hope that anybody who has made their own improvements to XCHANGE, to send me details, If you have made any substantial improvements and would like to help in the betterment of XCHANGE, we could supply you with the final product containing the changes without charge, but I'm sad to say not for everyone.

Please don't send me any disks or the actual changes to the program that you have made, just the details of what the changes do if Simon and I are interested we will definiteley get in touch with you. If and when the response is great enough I will pass the wish lists to Simon Goodwin and get the best ones incorporated then bring the new program into the community.

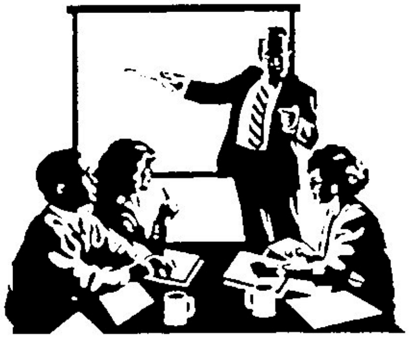
I cannot, if there is enough response, hope to reply to all the letters I receive unless there is an SAE and you are prepared to wait.

Mike Bedford-White, 16 Westfield Road, Acocks Green, Birmingham, Great Britain B27 7TL
Secretary of QL User Group (West Midlands).............(QUANTA sub group)

COMPUTERS 101 (A Tutorial) (Part 2)

London, ENGLAND - Mark Knight

3 Processors.



3.1 What the processor does; the instruction cycle.

At the heart of the functioning of every computer is a processor. The processor in action runs through an endless cycle of events, doing the same essential tasks over and over again.

First, a number is fetched from memory. This is treated as a code for one of the processor's instructions, so it is decoded, the instruction is executed, and then the next instruction is fetched. The processor normally continues to do this all the time it is switched on, endlessly fetching, decoding and then executing instructions until the power is switched off.

This fetch, decode, execute, fetch cycle is known as the INSTRUCTION CYCLE of the processor, or sometimes as the EXECUTION CYCLE. The details of how the instruction cycle works will vary from one processor type to another, and the precise meaning of each of the numbers will also vary. Some processors have a large, complex set of instructions, others have a smaller, simpler set.

The individual designs have their own collections of instructions, and the set for one particular processor is known as the INSTRUCTION SET for that processor. The set of numbers that make up the instruction set for a particular processor is the MACHINE CODE for that processor. All programs that run on a computer are ultimately decoded at some stage into machine code, otherwise the processor could not execute the instructions at all.

3.2 Registers.

On the processor chip, there are special memory locations that can be used to store numbers that the processor needs to use as data. These memory locations are known as REGISTERS. In order to add two numbers, most processors must have both numbers in registers on the processor, they cannot directly add up two numbers in memory.

To add two Words in RAM, therefore, the processor will usually have to execute three instructions at least. First, fetch the first number into a register, second, fetch the next number into another register, thirdly add the two registers. Depending upon the design of the processor, the answer may be in a special register for arithmetic answers, or it may be in one of the two registers that the original numbers were stored in. A forth instruction will often be used to move the answer from the register back into memory somewhere.

3.3 How many bits?

An old fashioned, 8-bit processor will usually have 8-bit registers, and often in order to use 16-bit numbers it will need to use a REGISTER PAIR, allowing two of the 8-bit registers to be combined for some instructions and used as if they were one register. Proper 16-bit processors will have true 16-bit registers, and can handle 16-bit numbers more quickly. Similarly, a 32-bit processor can handle 32-bit numbers efficiently and easily, and will normally be very much faster too.

The QL has an unusual processor, the MC68008, which is an 8-bit processor, as it has an 8-bit databus link to the main memory. What makes the MC68008 so unusual is that it has a 32-bit instruction set, and 32-bit registers. This makes it very efficient for an 8-bit processor, as it can readily handle numbers from single bits to 32-bit numbers using the built in instruction set. Fetching a 32-bit number from memory is still slower than with a true 32-bit processor, as it must be fetched in four 8-bit sections, but once inside the processor, the 32-bit value can be handled as one number.

The MC68008 is one of a series of processors, ranging from the 8-bit version used in the QL to some very fast, expensive 32-bit processors. The 7.5Mhz, 8-bit processor in the QL is replaced in the Gold Card by a 16Mhz, 16-bit version, the MC68000, so there is a faster system clock rate combined with the true 16-bit databus link to the Gold Card's fast memory chips.

This makes the QL fitted with a Gold Card run programs from two-and-a-half to over seven times faster than the unexpanded QL, with a little over four times faster being typical.

The 68000 range of processors using the same 32-bit instruction set, with the same registers on board. The range varies in the width of the databus, the size of the address bus, and in the speed and efficiency with which they can process instructions, even if run at the same clock rate. The 68000 variant is about twice as fast as the 68008 version if both are running at 8Mhz. The 68020, the slowest of the 32-bit versions, will run still faster at 8Mhz than the 68000. On the Super Gold Card, there is a 68020 running at a system clock rate of 24Mhz, which makes it much faster than even the Gold Card system.

The 68000 range is designed and made by Motorola, a large American company that specialises in making communications equipment, but is also one of the world's largest chip makers. The 68000 series are used in Apple Macintosh computers, in many UNIX systems, and as embedded controller chips in many telephone switchboards and other electronic equipment.

Another range of chips that are related to each other, the 80x86 series, is made by Intel, another large American corporation, the world's largest manufacturer of processor chips. This range is used in IBM Personal Computers, and in IBM compatible machines. The 8088 is the 8-bit member of the family, the 80286 is the 16-bit chip, while the 80386 is the first 32-bit version. The 80486 is a faster 32-bit system, and the Pentium, known as the 80586 in some circles, is an even faster, 32-bit chip with some 64-bit characteristics.

3.4 More on the buses - databus, address bus, expansion bus, local bus.

There is much confusion sometimes about what is meant by a 16-bit processor, a 32-bit processor or an 8-bit processor. As has already been stated, the databus is usually the deciding factor, so if a processor chip has a databus width of 16-bits, it is a 16-bit processor, if it has a 32-bit bus width, then it is a 32-bit processor, and so on.

Note that the address bus does not have to have the same width as the databus, and in fact usually it doesn't. The instruction set and registers, too, can be designed to have a different set of characteristics to those normally associated with the databus width for a particular processor. The QL's 8-bit processor is a good example, as it uses an 8-bit databus, a 20-bit address bus, but has a 32-bit instruction set, and so machine code programmers can treat it as a 32-bit system.

On some computers, there is another bus, known as the EXPANSION bus. This system is used to connect additional circuit boards in a standard form, known as EXPANSION CARDS, into the computer to improve its capabilities. Expansion cards can include graphics cards to improve the graphics capabilities, network cards in systems that do not have built-in networking, or sound cards to add better sound generation facilities.

A LOCAL bus is another kind of expansion bus, but connects devices more locally (in electronic terms) to the processor. This allows the processor to have more direct and rapid access to the additional devices on the expansion board. A traditional IBM PC expansion bus has a separate circuit between the processor and the expansion cards, so the access is slower.

A great deal of fuss has been made about the new local bus IBM PC systems, but much of this is catching up with what other systems have had for years. In fact, the QL has a local bus expansion port, where you plug in your Gold Card or Super Gold Card, if you are lucky enough to have one. The Apple Macintosh, Atari ST, Commodore Amiga and even the Sinclair Spectrum have local bus expansion systems. Local bus is just a fancy name for something that the IBM PC world has just discovered, while the rest of us have been using it all along.

3.5 RISC, CISC and vector processors.

During the relatively short history of computers, processors have become steadily more capable and much, much faster. Early processors were huge, filling a room, requiring special power supplies and made from thousands of separate components. Later, the MICROPROCESSOR was invented, etching all of the components onto a slice of silicon, the SILICON CHIP. This made the processor much smaller, and made it possible to run the resulting computer system from ordinary mains electricity.

COMPUTERS 101 - (CONT'D)

Early microprocessors were slow by current standards, and were usually 8-bit systems capable of handling 64k of memory, a mixture of RAM and ROM. Later, MEMORY PAGING was introduced, a nightmare for programmers but allowing more than 64k of RAM, by switching in and out 1k, 2k, 4k or 8k blocks, or "pages" of memory from a separate memory bank. Systems up to 4Mb were made this way, but they were a drudge for programmers, as it was the responsibility of a programmer to keep track of which page his code or data was stored on, and page it in and out of the switchable bank area.

Soon address bus widths grew, and databus widths, too. 16-bit processors with 20-bit, 24-bit and even greater widths to the address bus were made. Memory no longer needed to be switched in pages, and paging was, for the most part, happily left behind. Processors became steadily more complicated and capable, and designers concentrated on making them more and more powerful, with ever larger instruction sets to be decoded and executed with ever more speed and efficiency. The CISC, or Complex Instruction Set Computer, ruled the computer world.

One of the problems with this path of evolution is that the resulting chips became ever larger, demanding more and more power, and running at higher and higher temperatures. Some of the early Pentium PCs, for example, had serious overheating problems, as the processor chip alone consumes over 13 watts, and keeping such a small and power hungry component cool was a real problem for the computer designers.

In the mid 1980's, a new idea became fashionable among processor designers, the RISC, or Reduced Instruction Set Computer. RISC has several advantages over traditional CISC design, the first of them being the speed of the execution cycle. Since there are fewer, carefully designed instructions, it takes less time for a RISC chip to decode and execute each one. The chip, having fewer instructions, also needs less circuitry, and so it is smaller than an equivalent CISC design. The smaller amount of circuitry also consumes less power, is easier to debug during the design phases, and is much cheaper.

The disadvantage was that, to begin with, RISC processors were harder to program, as with fewer instructions to choose from a programmer had to think harder. In addition, where a traditional processor might possess an instruction to do something, a RISC chip might need several instructions to do the same job, and so sometimes RISC could lose much of its speed advantage. Progress in RISC design has largely eliminated these problems. Modern RISC chips now have instruction sets designed to be easier to use, and they process individual instructions so much faster than a CISC design that they are almost always considerably faster, even where a sequence of instructions is needed to do a task that one instruction could do on a CISC processor.

One feature common to many modern CISC and RISC designs is the inclusion of a built-in FLOATING POINT unit. This is like a separate processor on the same chip, specially designed to carry out floating point arithmetic using a specialised, small instruction set of its own. Where previously a programmer might have to write a large routine, possibly containing dozens or hundreds of instructions, to add floating point numbers, processors now tend to have a small number of instructions added to them to make this unnecessary.

As microcomputers have become more complex and sophisticated, the big, superfast MAINFRAMES have evolved, too. SUPERCOMPUTERS, the fastest machines used for scientific research and advanced design applications, are still changing. A feature of some of the fastest computer systems is called a VECTOR PROCESSOR.

The vector processor is a design that takes advantage of the fact that computer programs often need to do the same operation on many data items, say for example to multiply two hundred numbers by two. The vector instructions will mark a block of memory, and then pass the instruction to the processor to do the operation to all of the Words, Longwords or floating point numbers in the block. Instead of going through the numbers one at a time, as a traditional processor is forced to do, the vector processor can multiply all of the numbers by two (or by any other number) at the same time.

Vector instructions may use multiplication, addition, and the other arithmetic operations, as well as logical transformations and other types of instruction. The important thing is that all of the operations in the series are carried out at the same time, not one at a time in order. Some programming operations cannot be carried out in this way, for example if a series of calculations is required, and each depends upon the result of the previous one, then they must obviously be carried out in order, from first to last. The speed advantage, when the vector instructions can be used, is massive, and so it is still used in some mainframe designs.

3.6 Extra speed tricks - the pipeline, the cache, clock doubling and more.

During the execution cycle of a traditional processor, there are usually three main stages: Fetch, decode, then execute. While a processor is fetching an instruction, the decode and execute circuits lay idle, waiting for the fetch result. Next, the decode circuitry works, while fetch and execute lay idle, and finally the execution stage of the processor works while fetch and decode sit idle.

Somebody realised, a few years ago, that this was not necessary. If the circuits could be designed in a way that made them more independent of each other, then they could all be working at the same time. The first instruction fetched when the processor starts can be sent to the decode unit, and while it is being decoded the fetch unit can start fetching the next one. While the instruction is being executed, the second instruction has been fetched and can be decoded by the decode unit while the fetch circuit fetches a third.

This speedy system is now often employed on processors, and some take the idea much further, by employing super-fast fetch circuitry that can fill up with five, six or many more instructions. This can then feed instructions to the decode unit as fast as it can decode them, and then on to the execute stage as fast as this unit can work. This feature of a chip is called an INSTRUCTION PIPELINE, and it resembles the way that all the parts of a production line work at the same time.

Pipelines speed up many processors. There is a problem with a pipeline, though, because the fetch stage simply fetches instructions as fast as it can, one after the other. If a program contains a loop, then at the end of the loop will come an instruction to start fetching instructions from the start of the loop again, naturally. The trouble is, that when this instruction reaches the execution stage of the processor, the instruction behind it in the pipeline will come from after it in memory, not back at the start of the loop.

At this stage, the processor has to scrap the instructions in the pipeline and fill it from scratch. This is called FLUSHING the pipeline. The pipeline does give a speed advantage whenever instructions are processed in a linear fashion, but often jumps and loops in the machine code instructions will make flushing necessary, so the trick of keeping more than one part of the processor working at a time is stalled for a while. The problem arises even if the instruction to loop back is to loop back to an instruction that is already within the pipeline, as the pipeline is not location specific, so the processor does not know that it could just part-flush the pipeline and continue.

To supplement or replace the pipeline, some processors have a memory cache built in, not as a separate item but on the processor chip itself. This cache is filled in a similar way to a pipeline, but if an instruction loop fits entirely within the cache then a loop back instruction does not require the processor to fetch the instruction again, saving time. Instructions and data on some processors have their own cache areas, as the processor always knows internally if it is fetching data or an instruction.

The next big development in the never-ending quest to make computers faster was SUPERSCALAR PROCESSING. This can be designed into a processor along with a pipeline and a cache, and some other speed enhancements, too. Superscalar processing takes advantage of the same idea as a pipeline, that is, trying to keep areas of the processor chip working at the same time if they can be made independent. Instead of simply making different stages of the instruction cycle work at the same time, a superscalar chip will actually have more than one instruction cycle working at a time.

Usually, a superscalar processor has to have an on-chip memory cache, for instructions if not for data. This is because it may fetch more than one instruction at a time, and then feed the instructions to two separate execute stages. Suppose, for example, an instruction is to fetch a Word in memory into a register, while the next instruction is to fetch another Word, somewhere else in memory, into another register. These two instructions use different parts of the chip, and so a superscalar processor decode stage will detect the fact and send them for execution, by different parts of the execution circuitry, at the same time.

A superscalar processor may be, like the 68040, a 32-bit chip, yet the 68040 has a 128-bit internal databus for fetching from the on-chip instruction cache and pipeline. This allows it to fetch more than one instruction at a time, and permits superscalar operation. If two subsequent instructions use the same register, then the decode stage will mark them for sequential execution, and will pass them one at a time to the same execution unit, rather than trying to use both execution units together.

Note that a superscalar processor is not the same as a SCALEABLE PROCESSOR, which is another fashion among makers of RISC chips. The scaleable processor is one that has been specially designed so that it can easily be manufactured with different databus and address bus widths. So a 16-bit, a 32-bit and 64-bit version could be made, all capable of running the same programs, but costing different amounts of money and providing different speeds.

The simplest trick of all (and the last of those considered here to be invented), now often used to speed up processors started with what is called "clock-doubling". This uses the system clock pulse in an unusual way, dividing the pulse in two internally to generate a clock speed on the processor that is twice the clock speed of the rest of the computer system. If a processor is running in a 20Mhz computer, it will therefore be running internally at 40Mhz.

This trick makes less difference where a processor has no internal cache, as it will be limited to the speed of the rest of the system when reading instructions and data, and only the decode and execute cycles will be speeded up. The difference when reading from any internal pipeline or cache, is of course, dramatic, since in a clock-doubled system it will be able to proceed with a fetch twice as fast as a normal processor running in the same computer.

The clock doubled processor is not the limit, IBM and Intel currently both manufacture clock-tripled chips. Clock doubled 80486 chips are known as DX2 chips, and IBM clock-tripled chips are known as the DX3 chips. Confusingly, Intel sells clock tripled chips as its DX4 range, and some of the DX4 series from Intel are two-and-a-half multipliers, rather than doubled or tripled.

An 80486DX3/100 from IBM would run at 33Mhz externally, and 100Mhz internally, while an 80486DX4/75 from Intel runs at 33Mhz externally, but at 75Mhz inside the processor. Daft, I call it!

A note about clock multiplying and the QXL: All of Motorola's 68040 series chips are clock-doubled, so the 20Mhz 68EC040 used in the QXL card runs internally at 40Mhz - that's one reason why it is so fast.

Some readers may be wondering why clock-double at all - why not simply make a true 40Mhz computer, with all of the components running at that speed? Well there are three main reasons, the first two closely connected. First, cost; since all of the components will be required to run at 40Mhz, they will have to be made to cope with much faster signals, and so with finer manufacturing tolerances they naturally become more expensive. The second reason is that the system uses power at a rate that is partly dictated by the clock speed, so the power consumption will be raised by having the whole system running at this rate, which also causes more heat to be generated.

The third reason is that it is more difficult to design faster circuit boards, as they can give out radio frequency energy that interferes with televisions, radios and even other computers. Systems running at higher clock rates also become more prone to being affected by such things as strong radio signals and electromagnetic fields from other machinery. This means that designers would have to take longer to design such systems, and so we are back to cost; more money is required to buy them.

3.7 Parallel processing - the computers of the future.

Parallel processing takes the idea of a superscalar architecture even further, enabling computers to work still faster. In some of the best superscalar processors, up to six instructions can be carried out at any one time. The extension of this is to build a computer system with multiple processors, so each processor can carry out an instruction of its own. With ten processors, in theory the computer could be ten times faster than a computer with one identical processor, and with a hundred, it could be a hundred times faster.

In practice, the computer with ten processors turns out to be a little less than ten times as fast as another machine with one, because the processors must spend some time communicating in order to be able to work on the same task. Systems have, however, been designed with up to 65,536 processors, and these are, as you might expect, frighteningly fast, among the fastest supercomputers in the world. They also cost hundreds of thousands or even millions of dollars.

The expense of a massively parallel system is considerable, yet the number of such systems in use has exploded in recent years, because the old fashioned supercomputers that they replace used to cost hundreds of millions of dollars for a similar performance.

The very fastest, multiprocessor, vector processing supercomputers are still faster than the cheaper, microprocessor based, parallel systems, but nobody expects to pay hundreds of millions for a system that is fifty percent faster than another costing ten or twenty million, and so the older types will soon die out, particularly as the parallel systems become faster.

4 Input and Output devices.

In order for any computer, from a system taking up a desktop to one occupying an entire large building, to be of any use at all, it must have two things: A means of taking input from the users, so that they can persuade the system to do what they want, and a means of giving output to the users, so that they can use the results. The input device most commonly used at the moment is still the keyboard, while the most common output device is the screen.

4.1 Input devices - the keyboard, mouse, trackball, joystick etc.

Many input devices have one thing in common; they are SERIAL in nature. This means that they provide a stream of data in a fixed, unalterable sequence, which differs from a RANDOM ACCESS device. A random access device, like a disk drive, can read data in any order, for example in theory the data bytes on a floppy disk can be read in any order the computer may decide.

A keyboard provides the computer system with a stream of numbers, each one a code for the key or keys pressed by the user. The stream is in the order in which the user presses and releases the keys, and of course this order is not changed by the system. A mouse also provides a serial data stream, in the form of a series of pulses that can be interpreted as motion of the mouse, and so copied to the screen by moving a mouse pointer or other object.

A trackball works in exactly the same fashion as a mouse, and in fact it is often impossible for the system to tell the difference. Users, of course, know, they move the mouse, but the trackball stays in one place, and the ball is simply rotated in the desired direction of movement.

Joysticks sometimes work simply by switching on or off a set of small switches, indicating left, right, up or down, and the one or two "fire" buttons operate their own switches. This type of joystick is called a DIGITAL JOYSTICK, as the signals are single digits, 1 for on or 0 for off. The QL joystick ports are of this type.

An ANALOGUE JOYSTICK (ANALOG JOYSTICK for American readers) is rather different. It reports not just upon the direction of movement, but upon how far, so that a small movement to the left will send a different number to the computer than a large movement to the left.

4.2 Output devices - the screen, speaker, possible control ports.

The screen is the most obvious output device, and it is obvious, too, that it is not a serial device. The computer can alter the top, bottom or one side of the screen without affecting the rest, and can do so at any time. The screen is changed by altering the contents of a special area of memory, and the display chips then convert the contents of this memory into a video or monitor signal.

Often, the processor is concerned with changing the screen, but on some systems the main processor simply passes instructions to a GRAPHICS PROCESSOR, which makes the changes to the screen while the main processor does something else. Specialist graphics processors make the changes to the screen much faster than a general purpose processor could, and so speed up the system in two ways. The main processor can do something else while the graphics processor is changing the screen, and the changes themselves are made more quickly than the main processor could make them anyway.

Any sound devices, such as the humble (and feeble) QL speaker, are also output devices. A printer is an output device, usually connected to a parallel or serial port. Some computers possess control ports, for controlling heavy machinery or electrical or electronic equipment.

COMPUTERS 101 - (CONT'D)

4.3 Dual Input/Output devices - serial and parallel ports, networks etc.

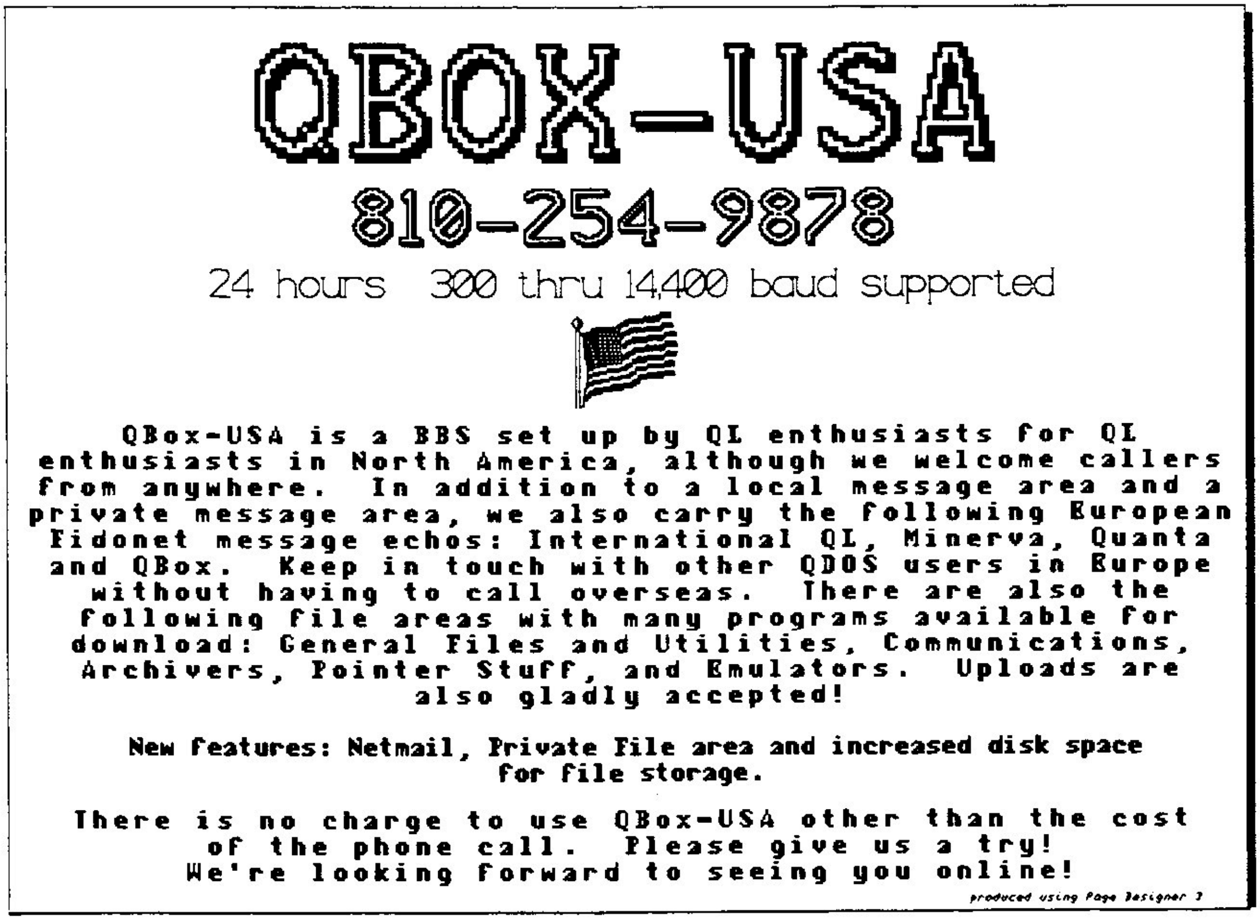
Some devices are both input and output devices, such as a serial port, which can be used, for example, to link two computers so that they can move data between them.

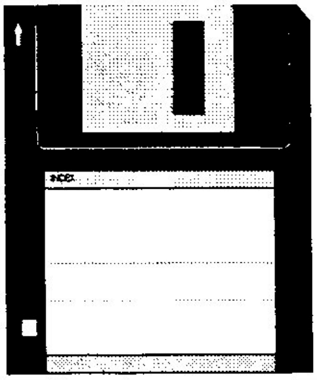
The serial port is a device to allow a variety of devices to be driven in a standard way, using electronic signals from the computer that represent a stream of bytes. The serial port switches a five volt signal on and off, at a predefined rate measured as the BAUD RATE.

The baud rate determines how long each signal will need to be to represent one bit, and these are grouped into bytes by the receiving device. The serial port can also receive data in the same way, at a rate set to match the device sending it. Two serial devices can only communicate if they can match baud rates.

The parallel port sends and receives data along eight lines at a time, making it possible to move a whole byte at once, or IN PARALLEL. This differs from a serial port where the bits are sent one at a time in a series, or IN SERIES. Strangely enough though, in one sense a parallel port is also a serial input/output device. This is true because although it moves bits in parallel, the bytes follow one another in a serial stream, just as the bits do through a serial port. Parallel ports can normally move data much faster than ordinary serial ports, because of their ability to move a byte at a time rather than a bit at a time. Specialised serial data lines do exist, however, and because a serial port is simpler electronically than a parallel port, transmission systems that have to be very fast are often serial in nature.

Such a specialist serial system is used in a NETWORK PORT, which allows the connection of a number of computers so that they can share a HARD DISK or some other storage device. Networks present special problems and advantages to computer users, and often networking circuitry is an expensive add-on to a computer. The QL has a relatively slow but quite usable network facility built in, and with some additional software (SuperToolkit II) it can be a very effective network system.





sinclair

£ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £

QLerk

A FINANCIAL PROGRAM FOR THE SINCLAIR QL

I'll bet you don't keep your financial information on your QL or if you do you only keep part of it and most likely in a Spread Sheet format. QL users haven't had much of a choice up until now. With QLerk you now have the capability of keeping complete and accurate personal and small business financial information on your QL. A friendly software version of a Financial Clerk is ready to serve you. The program is written in the ARCHIVE Programming Language. You might think ARCHIVE is not a 'real' enough environment to handle a complex financial system. Try the DQLerk demo system and see for yourself. And since it was done in ARCHIVE it is an open system not hidden behind a compiler. Everything is completely accessible and changeable by the user. Suppose the printout of a check (cheque) is not exactly right for your situation and you know a little ARCHIVE. Just load the program, bring up the ARCHIVE editor and change a few lines in the 'print_check2' procedure. Try doing that with any other financial program. If you can't do it then I'll do it for you for $10.

All functions are selected from standardized menus. No knowledge of ARCHIVE is required. The program works using common sense rather than imposing an accounting methodology on the user. Although it has many powerful features the user can use only those features desired, ignoring the rest. Recommended minimum system is a Trump Card with 2 DD Drives. It works much faster on a Gold Card or Super Gold Card or QXL. It multitasks because it runs within ARCHIVE which can be multitasked. It has an on-line help system which can be viewed, searched, and printed from most menus. At present it does not work with XCHANGE ARCHIVE. A copy of ARCHIVE is furnished with the program.

It is international in the sense that it lets the user set the currency and date entry style and the addresses are in an international format. Many of the fields are set up as templates so the user has ultimate control of what they are called. It has been developed carefully over the past 4 years in real world situations (current version is 3.21).

Pays Bills
Makes Invoices
Prints Address Labels
Does Payroll
Does Periodic (Cyclic) Payments
Maintains Cash Accounts
Elaborate Data Protection
Category Report (why,when)
Will Automatically Set Up RAM Files

Receives Income
Makes Purchase Orders
Handles Sales Taxes
Keeps Inventory
Does Periodic (Cyclic) Income
Maintains Charge Accounts
Easy Built-in Data Backups
Activity Report (who,when)
User Definable Fields

Reconciles Bank Accounts
Prints Checks
Handles Income Taxes
Handles Periodic Payment
Maintains Savings Accounts
On-line Help
Easy Data Correction
Powerful Address Subsystem
and much more

PRICING

DQLerk (a Public Domain Demonstration Version of QLerk)	$5.00 US/Canada	$7.00 Elsewhere	(refundable if QLerk ordered at a later time)
QLerk Program on Disk with Tutorial Doc File	$29.00 US/Canada	$31.00 Elsewhere	
QLerk Manual (150 pages of details)	$29.00 US/Canada	$34.00 Elsewhere	
QLerk Program with Tutorial and QLerk Manual both	$50.00 US/Canada	$57.00 Elsewhere	

The above pricing is for payment in US Dollars by Postal or Bank or Travelers Check with Wood and Wind computing as the payee. Personal checks in US Dollars okay from US/Canada. I do not accept credit cards but I have made arrangements so that I can receive payment by personal check as long as the check is in the standard currency of the country in which the bank resides. So I can accept personal checks in Pounds from the UK and so forth. I must add an extra charge for my costs for this service :

For the UK		For Germany		Elsewhere	
QLerk Program	28 Pounds	QLerk Program	67 DM	QLerk Program	$46 in your currency
QLerk Manual	30 Pounds	QLerk Manual	71 DM	QLerk Manual	$49 in your currency
Both together	46 Pounds	Both together	107 DM	Both together	$72 in your currency

Postage is included in the price and all orders will be shipped immediately by air mail. DQLerk (the Demo version) will show you many of QLerk's features so you can decide if you really want the program but it is not suitable for keeping real financial information. You may find a copy of DQLerk in your users group library or on a bulletin board or in the QUANTA Library. The QLerk Program comes complete on a disk with a Tutorial Doc file. The most commonly needed features are covered by the Tutorial and those with simple needs would probably find this sufficient. There is also a complete indexed manual of 150 pages which I offer separately or together. Those with complex finances or a desire to know all the details should purchase the Program and Manual both. Special functions and reports can be created upon request for a small fee. **So Increase the usefulness of your QL dramatically by ordering QLerk today from :**

Wood and Wind Computing
Bill Cable
RR3 Box 92
Cornish NH 03745 USA

Phone : (603) 675-2218

£ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £ ITL $ BEF DM £

DJC **DILWYN JONES COMPUTING**

41 BRO EMRYS, TAL-Y-BONT, BANGOR, GWYNEDD, LL57 3YT,
GREAT BRITAIN
TEL+FAX: (+44) 01248-354023

WINBACK £25.00

Compatible with the recently released QUBBIDE IDE hard disk interface from Qubbesoft, a good quality backup program is essential for the hard disk owning QL user. Look at these features:

- Incremental backups. Can copy contents of sub-directories.
- Ability to split large files into sections over several disks.
- Backup only files changed since last backup (saves time), or all files.
- Listing options - what went on which disk?
- Utility to rebuild your directory structure after a disk crash.

THE SPECIAL OFFERS OPPOSITE...

As a special offer from now until the end of 1994, I have decided to reduce the price of some software, see the offers marked on page opposite. The five programs are:

BANTER. A banner and text poster making program. Very simple to use, choice of fonts and compatible with 9 and 24 pin dot matrix Epson compatible printers.

IMAGE PROCESSOR. Touch up existing graphics or use as a graphics and drawing package in its own right. New V2.08 now adds facility to save in pointer environment area save bitmap format. Upgrade earlier versions for only £3.00.

SIDEWRITER. Print wide spreadsheets or plain text files sideways on a dot matrix printer, with a choice of text sizes.

SOLITAIRE. Enjoyable Klondyke Patience card game, price reduced by a huge 33% !!

TEXT 'N' GRAPHIX. A program to help you include QL screen pictures in printouts of Quill text files.

THE POINTER ENVIRONMENT & QPAC2

POINTER ENVIRONMENT "IDIOT'S GUIDE" You may have heard a lot recently about the Pointer Environment and QPAC2, but may have been frightened off by reports that it is difficult to learn and to set up. True, there is a lot to learn, and this can make it seem difficult at first. But DJC has listened to QPAC2 users and done something about it! A new 20 page booklet by Norman Dunbar sets out to provide a simpler introduction to this most essential of QL software. Pointer Environment will become an essential part of every QL user's software setup sooner or later, especially when enhanced QL graphics hardware becomes available! Oh, the title - don't be offended, all is explained in this booklet! Available FREE on request with orders for QPAC2, or £2.00 to cover costs if you already have QPAC2.

QPACCER - Shareware program to help you to set up boot programs for use with QPAC2. Expanded memory required. QPACCER is available from QL PD libraries, or from DJC for £2.00

QPAC2 .... £39.95 QPAC1 ... £19.95

QPAC1 is an accessory pack of several small pointer driven programs which can be used with or without QPAC2

LINE DESIGN PROGS

It was used to produce and print this ad - say no more! £100.00
(printed from a QL using a HP Deskjet 500)

PROFORMA FONTPACK

ALSO AVAILABLE - new pack of 100 high quality fonts for use with LineDesign and Proforma £80.00

PROFORMA

'C' programmers can now access the drivers, graphics etc of Line Design! £100.00

DATA DESIGN database £60.00
DATA DESIGN API £20.00
PFData £20.00 PFlist £20.00
pfb2pff (font converter) £60.00

TEXT 87 PLUS 4 £79.00

Powerful QL wordprocessor. Proportional spacing, choice of fonts, rulers, frames and columns for text and much more.

2488 Printer drivers for 24 pin dot matrix and bubblejet printers £29.00

TYPESET-94 Printer drivers for HP Deskjet and laser printers. £29.00

FOUNTEXT94+FOUNTED89. Graphic mode driver and founts. £39.00

PUBLISHERS PACK. Powerful link between Text87 and Line Design allowing transfer of formatted text and layout information, giving extremely powerful DTP capabilities. £199.00, incl. Line Design.

GENEALOGIST 3 £60.00

Latest version of the best-selling Genealogist software by Chris Boutal, now pointer driven and even more facilities. Makes family history and family trees a delight on the QL!

UPGRADE FROM 2ND ED. £33.00
QL GENEALOGIST 2ND ED. £30.00
GENEALOGIST 128k £12.00
Version for unexpanded QL.

DJC DILWYN JONES COMPUTING

41 BRO EMRYS, TAL-Y-BONT, BANGOR, GWYNEDD, LL57 3YT,
GREAT BRITAIN. TEL+FAX: (+44) 01248-354023

Address Book & Label Printer		15.00
AMD Airplan		10.00
Banter	**SAVE £5.00**	20.00
BASIC Reporter		10.00
Bible text disks		20.00
Cocktails Waiter		10.00
Convert-PCX		10.00
Cricket Secretary		12.00
Cue-Shell		40.00
Data Design 3		60.00
Data Design API		20.00
DBEasy		15.00
DBProgs		15.00
DEA disassembler		26.00
Deskjet-A5		12.00
Disa 2		40.00
Discover		20.00
DJToolkit		10.00
Easyptr 3 part 1		40.50
Easyptr 3 part 2		20.00
Easyptr 3 part 3		20.00
Filemaster		12.00
Files 2		12.00
Five Game Pack		12.50
Flashback		25.00
Flashback Special Edition		40.00
Fleet Tactical Command		39.95
Flightdeck		15.00
Floppy Disk Utilities		18.00
4Matter & Locksmithe		23.50
Fountext94+Founted89		39.00
(Text87 driver)		
FTC Data Printer		9.95
Fugitive		9.95
Genealogist 3		60.00
Genealogist 2nd Edition		30.00
Genealogist 2 to 3 upgrade		33.00
Genealogist 128k		12.00
(budget version)		
Grey Wolf		12.50
Home Budget		20.00
Image-D		10.00
Image Processor	**SAVE £3.00**	12.00
Line Design 2		100.00
Locksmithe		14.95
Masterbasic		22.00
MDV Toolchest		14.95
MegaToolkit (disk)		25.00
MegaToolkit (eprom)		40.00
Micro Process Controller		64.50
MPC software		9.95
(Toolkit for use with Process Controller)		
Multi Discover		30.00

Music Manager		12.00
Network Prover		4.00
OPD Interchange		15.00
Open Golf		12.50
Open World		18.00
Page Designer 3		40.00
Painter		25.00
P.A.Y.E. Master (UK only)		30.00
PD2 Clipart		10.00
Picturemaster		15.00
Picturemaster Plus		20.00
Printermaster		20.00
Publishers Pack		199.00
QIndex		20.00
QL-PC Fileserver		24.50
QLiberator 3.36		50.00
QLiberator, budget version		25.00
QLibrary Manager		18.00
QLOAD and QREF		15.00
QPAC1		19.95
QPAC2		39.95
QPower regulator		24.95
QRactal		20.00
QTOP		29.95
QTYP2		29.95
Question Master		10.00
Quick Mandelbrot		15.00
Quick Posters		10.00
Quiz Master		10.00
Ramdisk		2.00
Return To Eden		17.50
Rob Roy Pack		10.00
R.P.M.		15.00
Scanned Clipart 1		10.00
Scanned Clipart 2		10.00
Screen Compression		10.00
Screen Dazzler		15.00
Screen Economiser		10.00
Screen Snatcher		10.00
S_Edit editor		20.00
SerMouse (+2.50 postage)		40.00
Sidewinder Plus		24.95
(screen dumps)		
Sidewriter	**SAVE £3.00**	12.00
(sideways text printing)		
Slowgold		5.00
Solitaire	**SAVE £5.00**	10.00
Save 1/3 on normal price!		
Spectrum Emulator Zm/128		28.00
Spectrum Emulator Zm/HT		40.00
Speedscreen (disk/mdv)		15.00
Speedscreen		
(Eprom+disk/mdv copy)		30.00

Spellbound	30.00
Spellbound Special Edition	50.00
Squidgy Round The World	12.50
SToQL	12.50
Super Disk Index	12.00
Super Disk Labeller	10.00
Taskmaster	25.00
Taskmaster hard disk upgrade	5.00
Text87 Plus 4	79.00
2488 Text87 drivers	29.00
Textidy	15.00
Text 'N' Graphix	15.00
The Clipart	12.00
The Gopher	12.00
3D Terrain	12.50
Toolkit 2 (disk/mdv)	19.95
Toolkit 2 (EPROM)	24.95
Trans24	10.00
Typeset94 (Text87 drivers)	29.00
Video Titles	5.00
Vision Mixer 1	10.00
Vision Mixer Plus	22.50
Winback	25.00

SPECIAL OFFERS VALID TO 31/12/94

FLOPPY DISKS & LABELS

FLOPPY DISKS DSDD 3.5"	0.40
FLOPPY DISKS DSHD 3.5"	0.70
ED DISKS - Call to ask for prices!	
DISK LABELS 3.5" roll 100	2.00
DISK LABELS 100 on tractor feed	2.50
ADDRESS LABELS roll 100	2.00
MICRODRIVE CARTRIDGES	2.50

Add 2.50 postage for floppy disks, or 0.50 postage if only ordering labels.

MAGAZINES

QReview magazine UK per issue	2.00
QReview (Europe)	2.50
QReview (other countries)	3.75
IQLR from Vol 4 Issue 2 (each)	4.25

CATALOGUE

Call or write for a FREE copy of our full 28 page QL catalogue.

DISCOUNTS (SOFTWARE ONLY) 5% FOR 2 PROGRAMS, 10% FOR 3 OR MORE,INCLUDING OFFERS!

TERMS AND CONDITIONS. All prices shown in UK Pounds Sterling. Software sent post free to U.K., overseas add 1.00 per program (maximum 3.00). Bulky items - see above for postage. PAYMENT - Make cheques or Eurocheques (in Pounds Sterling, drawn on UK branch of bank or building society) payable to "Dilwyn Jones Computing". Payment by credit card (Visa/Access/Mastercard/Eurocard) also accepted. Minimum order is now 5.00 due to bank charges. Goods, whether individually identifiable or not, remain the property of Dilwyn Jones Computing until payment in full for them has been received by DJC. Under credit card company rules, we can only send goods paid for by credit card to the cardholder's registered address.

VISA
ACCESS
M'CARD
EUROCARD
all accepted!

10 and STILL GOING !!!

Trafford, Alabama, USA - Bob Madaris

So the QL was ten years old this year. Well it looks more like it 's growing like a new born babe.

This QL of mine is seven years old and has the original keyboard (one membrane change). For the first 2 years I used it unexpanded to do most of the things I wanted it for. I quit using the TS 2068 (similar to the Spectrum) and the ZX-81 altogether. I built a cable to use the Magavox RGB 80 monitor that I had used with the TS 2068. I finally decided that I needed more memory and floppy drives. I added a 256k TRUMP CARD and dual 3.5 inch floppy drives (720k naturally). I later learned that this Card could be upgraded to a full 768k by installing Drams. These were obtained from Ads in Computer Shopper magazine and then I had a full 896K. All the programs that I had been using up to this time, could easily be handled by this combination. I was using a 9 pin dot matrix Sears SR2000 printer. (Actually to start with I was using a Gorilla Banana).

MIRACLE SYSTEMS came out with the GOLD CARD, but for sometime I thought I just really did not need it. I finally gave in to my craving and ordered it from Mechanical Affinity. Wow what an improvement. I even let Bob Dyl twist my arm in doing an occasional review and article for IQLR after which I upgraded to the HP DeskJet 500 printer.

Back at the time when the membrane gave me trouble, I was coming up on a deadline for estimated Federal income tax and all the information was kept and computed in ABACUS. I had learned to backup all my programs from the many articles printed on the subject. But, I did not have a backup QL. A frantic call to RMG and for $75, I had a QL in time to get my tax in on schedule. I then sent my sick QL for a membrane change.

Now with the Gold Card on my main QL and the full Trump Card on the backup, I'm able to do some networking between the two. I use the drives and printer of the main system when I'm networking. I did add a QL Vision monitor to the backup system when Mechanical Affinity had a sale on used monitors.

In my seventh year, (the QL's tenth) MIRACLE SYSTEMS came out with the SUPER GOLD CARD. AW, I don't need it, said I. There are other things that were on my wish list.

John Impellizzeri and Don Walterman wrote articles in IQLR on going online with the QL. I had obtained some disks from Paul Holmgren that had been downloaded from a BBS. This is what I would like to spend my next money on. But first I obtained and installed the HERMES chip, as I knew I had to have it. I placed an order to Midwest Micro and obtained a USRobotics Sporster Fax/Modem (thanks for the tip on this modem in IQLR). How to make the cable was in an the article by John in IQLR (thanks John & Don). I got online using QTPI with Q-BOX USA but I am still learning. I downloaded a program into Ram that was too big for my floppies. I just have to get the ED drives. But now I needed the Parallel port on the SUPER GOLD CARD for the printer. So, I ordered and obtained a SUPER GOLD CARD from Frank at Mechanical Affinity (traded in the GOLD). Wow, what another big difference and a true parallel port for the printer.

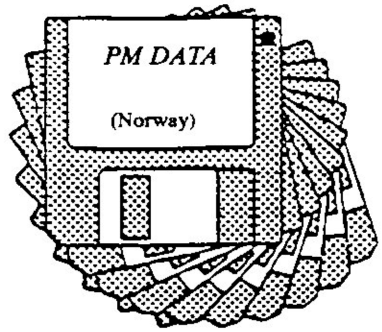
Oh well, I had to have the ED drives, so I ordered a pair of TEAC drives from MidWest Micro and with the kind assistance of our publisher got the jumpers corrected, added cable and all put together in a case with power supply. Gosh, 6400 sectors, I would never have believed that. Now I don't have to boot from both drives. I will be able to UNZIP those files to Ram and copy them to floppy when I download the big ones. What's next for me hardware wise? The new IDE Interface and a hard drive? Keyboard interface and a 101 keyboard? Put it all in a tower case (Bob Dyl gently hints to clear up all my clutter)? The GRAPHICS card? Whatever?, it will be , Ill use my QL for as long as IT and I are still able to compute!

Thumbnail profile of this writer:

I am 67 years of age and retired. I spend most of my evenings at the QL. My working career was mainly in Aircraft Electrical, Electronics, and Avionics maintenance and repair. I am not a programmer. I take other's programs and piece portions of this one and that one to suit my needs or imagination. I do a lot of what if's, all that can happen is to crash the computer. NO PROBLEM.

DISK MATE 4 (Version 4.45)

Hamburg, GERMANY - Roy Wood



One area of programming that is especially well represented on the QL is file handling. This is hardly surprising really since all information held by a computer is in a file of some kind and the one thing you learn early on is that making backups stops headaches and large dents in the wall nearest to the computer. Although the system itself has a large selection of file handling commands most of us are much happier to be able to call up a program which gives us the various options on screen and presents directories in an easy to assimilate form.

I have three different file applications on my QL. The first is, of course, QPAC 2 almost standard equipment these days. For those of you who have not upgraded to the latest version of this program it has a very neat little trick. If you have QD6 LRESPR'd and you click the execute command from the QPAC directory on a file that is not executable it invokes QD and displays the file ready to be edited or viewed using QD's excellent facilities.) I also have Albin Hessler's Cueshell and this has a lot of very useful applications but Disk Mate 4 (DM4) has some functions that neither of these have and is, in my view, well worth investigating.

I first came across this program at the International QL meeting in Bielefeld, Germany in February of this year. I sat down at the NASA (Norwegian All Sinclair Association) stand and had the program demonstrated to me and I bought it immediately. (NASA seems to have become PM DATA now but I have seen no explanation of this) This program started life some time ago as Disk Mate but this did not run under the pointer environment. This version, as the manual states in the first paragraph, is completely rewritten and not only sports the full pointer facilities but also has little in common with its predecessor except the name.

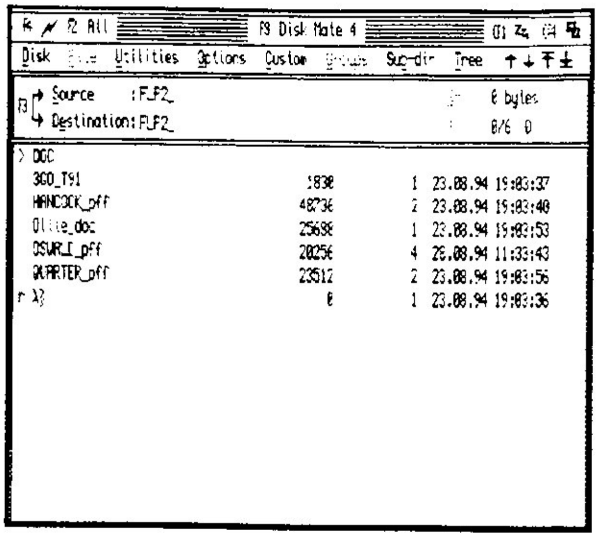
I was very impressed with the demonstration but when I got it home I found there were a couple of problems with my particular setup. Pål Monstad, the author of the program, proved to be very diligent in rectifying them and the version that I now have works perfectly. This is not a real criticism because, as I have said elsewhere, the QL has developed and diversified so much in its ten year history that small incompatibilities between the different CPUs can be big pitfalls for the programmer.

The program comes on one 3.5" DD disk and is accompanied by a concise A5 manual. The system requirements are at least 512k of memory, a 3.5" disk drive, Toolkit II and level 2 device drivers.The problems I had with versions 4.42 / 3 seemed to be either Gold Card or Minerva incompatiblities and in his letters to me Pål said that he developed this program on an Atari TT with the QVME emulator - "the best computer I have ever owned" (quote). I ran this version on a JS QL with the Super Gold Card v 2.49, Minerva V 1.97 and the Hermes chip installed and encountered no problems. Since it is a pointer environment program it comes complete with the standard files that are needed to run it and with Jochen Merz's Menu_rext. It can be configured to your particular system by the usual 'config' program or by Jochen's Menuconfig (not supplied but I find this much easier to use than the standard config).The program itself is in Qliberated Superbasic and comes in two versions, with or without linked runtimes. This saves space on the computer (10k) for those who already have the Qliberator code running. DM4 also needs a few toolkit commands to be loaded and these are supplied in a file called "extensions_cde " which can be LRESPR'd in the boot file.

DM4 can be started from the supplied BOOT file or, if you have already got the pointer files, Menu_rext and extensions_cde loaded, by issuing a simple EX or EXEC command. The program can also be made resident in the usual QPAC 2 manner and given a HOTKEY and a 'button'(this is how I use mine).

When the program is started you are presented with a pleasant dark green / black screen (fig1) the top line of which gives the familiar pointer symbols for redrawing the screen, putting the program to sleep or moving the window (although if you have the standard QL display then this will be of no use). In addition to this you can press F9 or hit the Disk Mate 4 name to get the version number, serial number and company address or press F2 or hit the All box to select all the files on the directory. The next line gives you a series of pull down menus for the various functions that the program will perform. These can either be hit with the mouse or selected by the underscored letter although some are not available until either a directory is loaded or a file is selected. At the end of this line are four arrows the first and second of which move the screen up or down one page and the last two move to the beginning or the end of the directory.

DISK MATE 4 - (CONT'D)

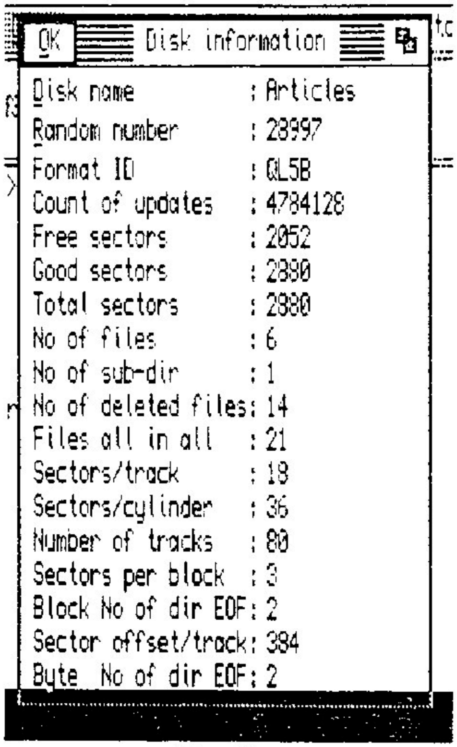


```
Source      : FLP2_                 0 bytes
Destination: FLP2_                  0/6  0
> DOC
  3GO_T91              1838    1  23.08.94 19:03:37
  HANCOCK_pff         48736    2  23.08.94 19:03:40
  Ollie_doc           25698    1  23.08.94 19:03:53
  OSWRLI_pff          20256    4  26.08.94 11:33:43
  QUARTER_pff         23512    2  23.08.94 19:03:56
r X3                      0    1  23.08.94 19:03:36
```

(Fig. 1)

The next Section of the display shows the Source and Destination Devices for the files and these two can be swapped by pressing F3 or changed by a hit on the device name thus invoking the Menu_rext screen. These two functions take their devices from the 'PROG_USE', 'DATA_USE' and 'DEST_USE ' commands in your Boot file and if these are omitted the destination becomes the printer port. Care needs to be taken to return the source and destination to the default state before you leave DM4 because changing them changes the system defaults too.This can either be a feature or a curse depending on how you use your machine . On the far left of this section are two info lines which tell you how many files are selected out of the total and the number of bytes they occupy and the arrows for moving back and forth through the disk's directory tree. Below this section is the main window for the display of the files.This is bounded by scroll arrows at the top and bottom and has the usual scroll bar along the right hand side.

The "Disk" menu gives access to the following functions: Directory, Format, Disk info, Write Directory, Sector Copy Disk, and Quit. The first two of these are obvious although the Format command gives the added instruction to format a disk to a set capacity which will allow the user to, format a HD disk to a DD capacity that will be readable on a DD disk drive. Disk info is a very comprehensive list of the disk's contents (see fig 2) and gives the user the ability to simply change the disk's name and number. Write Directory will allow you to write the directory back to the disk exactly as it appears on the screen. This can be a very useful function because later parts of the program allow you to resort the directory and group certain files together and all of this is retained when the directory is written back. This function does not work if directories are present on the disk and will only work on DD disks.



```
Disk name          : Articles
Random number      : 28997
Format ID          : QL5B
Count of updates   : 4784128
Free sectors       : 2052
Good sectors       : 2880
Total sectors      : 2880
No of files        : 6
No of sub-dir      : 1
No of deleted files: 14
Files all in all   : 21
Sectors/track      : 18
Sectors/cylinder   : 36
Number of tracks   : 80
Sectors per block  : 3
Block No of dir EOF: 2
Sector offset/track: 384
Byte  No of dir EOF: 2
```

(Fig. 2)

The "File" Menu is not available until a file has been chosen from the directory but then will give the usual copy/delete/rename/view options familiar to most file programs. The rename function of this menu, however, allows you to stack up a pile of renames and then perform just one writing to disk operation - very useful if you want to change a whole batch of different files into a lot of different names.

Selecting the Rename option from the this menu will give you another menu consisting of Rename, Replace, Erase (fig3). Replace will replace any occurrence of a string in the file name with any other string and allow you to set the number of times that it will do it. (The example in the manual is replacing the every occurrence of "a" in a file called "alabama" to "test". With the maximum number of changes set to 1 it becomes "testlabama" set to 4 it is "testltestbtestmtest") Erase works in the same way except that it erases the string occurrences. Copying, can be done in two different ways too; Standard is the normal slow method but this is useful if you have only limited memory in the machine, when there is memory to spare the fast copy method can be utilised.

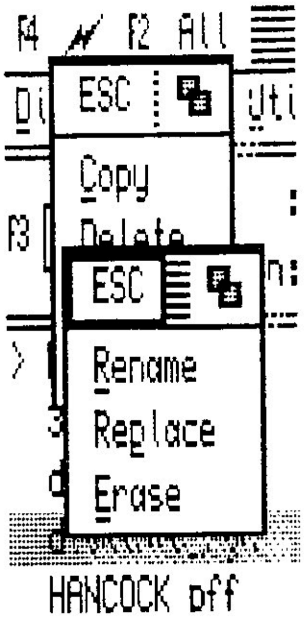
The Utilities menu enables you to do three distinct things. The first is to alter the data space in Executable files but there is a warning in the manual not do do this unless you know what you are doing! The second option will enable you to convert strings within programs. This is amazingly useful because it works not only on Superbasic programs but also on machine code and data files! (I converted my Psion Chess to run from a floppy disk by using this) The last option will print the directory for you.

DISK MATE 4 - (CONT'D)

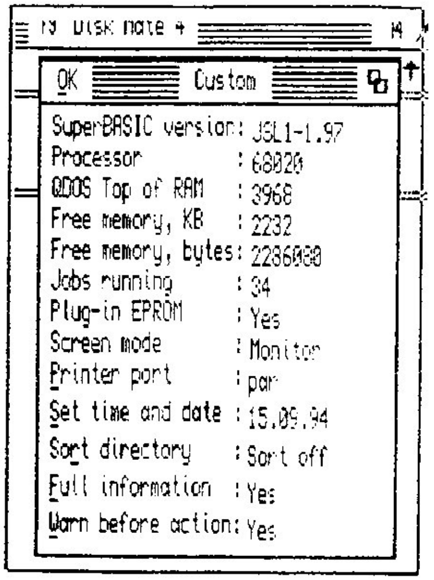
The Options menu will perform the usual file sorting functions with the addition of sorting by length and dataspace it will also change the case and select by name (wildcards included) or by date. The Custom menu will tell you all that you need to know about the QL (fig4).

The Groups menu allows you put files into groups and then sort the file by these criteria again a useful facility for disks that have a lot of files on them.The last two options are Sub_dir and Tree and allow the making and deleting of sub-directories and the expanding of the file info into its full form.

This is a somewhat long review because this is a very comprehensive program and one which continues to prove useful in unexpected ways. Pål is currently working on a command to copy complete subdirectories and I am sure that other improvements will follow. I have suggested to him that a configurable source and destination would be a better option because of the chance of sending a file somewhere you did not expect as a result of inadvertently leaving the defaults in a different state to the one that you had originally. He is very open to suggestions from users and always does his best to rectify any faults that turn up. As I said at the beginning there are many file programs around but if you want the "All singing all dancing" version then this one is well worth having.



(Fig. 3)



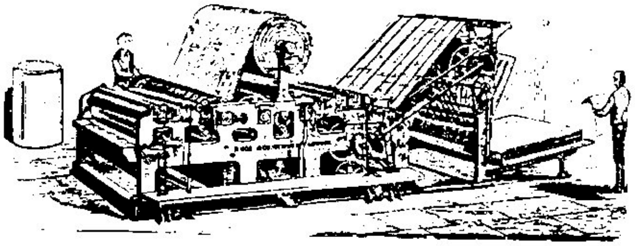
```
SuperBASIC version: JSL1-1.97
Processor         : 68020
QDOS Top of RAM   : 3968
Free memory, KB   : 2232
Free memory, bytes: 2286080
Jobs running      : 34
Plug-in EPROM     : Yes
Screen mode       : Monitor
Printer port      : par
Set time and date : 15.09.94
Sort directory    : Sort off
Full information  : Yes
Warn before action: Yes
```

(Fig. 4)

Disk Mate 4 is available from :

PM Data,
Nerheim
N5580 Ølen
Noway

The cost is 390,- NOK (Norwegian Kronor), or 90,- Dm (Deutschemark)You can pay by Eurocheques made out in Kronor or American express Travellers cheques ($66,-) The price includes postage by AirMail Worldwide.

STOP the 

After I had written the above article I received version 4.48 of Disk Mate 4 from Pål. This, as it says in the updates file on the disk is a major reworking of the program and incorporates a suggestion of mine - namely that whole subdirectories can now be copied. I have not really had the time to thoroughly test the program but it seems to run faster and smoother than previous versions and there are a few provisions for further development that show that PM Data is committed to a policy of upgrades. Reading the updates_txt file on the disk will give you a good idea of the amount of work that has gone into this program.

One of the main points is that the subdirectory option has gone from the main menu and now resides in the files menu and its place has been taken by a new menu item - Macro. In this version that item has not been implemented but the letter that accompanied the disk suggests that it soon would be. Another innovation is that the Custom Menu now gives the SuperBasic version and the QDOS version on the first line (in my case JSL1-1.97) and the Processor on the next (68020 for the Super Gold Card). All the problems that had occurred in earlier versions with full HD disks and clocks that were wrongly set have been cured.

NEW Telephone/Fax 0494 871319 (EEC) W.N. Richardson & Co.

SINCLAIR QL PRICE LIST AUGUST '94

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire, SL9 0NB

Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88

WE NOW STOCK CAMBRIDGE Z88 PORTABLE AT £99 AND CARRY A FULL RANGE OF ACCESSORIES

Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88 Z88

S.A.E. for further info. on hard disk drives, floppy disk drives, Z88 and Falkenburg products and accessories

BACKUP QL WITH P.S.U. JS ROM - £80 JM ROM - £70
PART EXCHANGE ALLOWANCE - £15

Accessories

* NOTE: EXTERNAL (SER2) 3 BUTTON MOUSE AND SOFTWARE WITH EXTRA FUNCTIONS. NOW HERMES COMPATIBLE. *

PC KEYBOARD INTERFACE, for INTERNAL FITTING, with lead to 102 KEY KEYBOARD	£ 75.00	
PC KEYBOARD, UK version, 102 key (AT) — REDUCED PRICES →	£ 25.00	~~£ 30.00~~
* PC KEYBOARD INTERFACE and KEYBOARD —	£ 95.00	~~£ 99.00~~
CASE and LEAD for EXTERNAL FITTING of Keyboard Interface for simpler assembly	£ 14.00	
SER MOUSE, 3 Button, Software controlled, with operating and formatting disks externally mounted, fits in SER2	£ 40.00	

TANDATA MODEMS AGAIN IN STOCK

1Mb, 2Mb and 4Mb

ECONOMY MULTI-COMPUTER FLOPPY DISK DRIVE

NB: 4MB DRIVES NEED GOLD CARD OR SUPER GOLD CARD

CAPACITY:	1Mb DS/DD	2Mb DS/HD	4Mb DS/EHD
SINGLE DRIVE	£69	£85	£120
TWIN DRIVES	£99	£150	£199
10 VERBATIM DISKS	£5	£8	£35

TELEPHONE RE: UPDATING EXISTING DRIVES. ALSO DETAILS OF HARD DISK DRIVES.

Monitors

PLEASE ENQUIRE FOR COLOUR AND MONO MONITORS £40-£150

Microdrive Cartridges and Spares

4 New Cartridges in a wallet £ 10.00 8 prog carts for reformatting in 2 wallets £ 15.00
Plastic Storage filing box including 20 new cartridges £ 45.00
QL Psion Software V2.35 Includes Quill, Abacus, Archive, and Easel IN WALLET £ 18.00
QL Psion Software Separate programs £ 10.00

QL power Supply Unit £ 10.00 QL Printer I/F £27 Membrane (and instructions) £ 9.00
TV or Network leads £ 3.00 QL Top & Bottom Case £ 5.00
IC's ZX 8301 £ 9.00 ZX 8302 £ 3.00 8049 (IPC) £ 3.00 MC 1377 £ 3.00

QL SERVICE MANUAL & CIRCUITS £25.00 S.A.E. FOR FURTHER DETAILS

VISA

Payment terms:
CWO, Access, VISA et cetera
Cheques - allow 10 days

Delivery:
Carriage - £9
Postage - £5

Access

Product subject to availability, E&OE.

TEL/FAX: 0494-87-1319

MOBILE:
0850 597650

SBASIC

Newmachar, Aberdeen, SCOTLAND - Norman Dunbar

'Great News ! The enclosed disc contains SBASIC' the letter from Miracle Systems began.

The disc in question was release 2.16 of the QXL software and after backing up the disc and trying it out, it was indeed impressive.

As outlined in the article in IQLR volume 4, issue 2, by Thomas Robbins, the SuperBasic interpreter on the QXL was pretty limited - well actually very limited indeed - no line numbers, loops, expressions etc.

With release 2.16, all this has changed, and changed for the better - SBASIC has arrived and is a multitasking version of SuperBasic for the QXL written by Tony Tebby and in my opinion, a great job has been done.

As yet, there is no manual to speak of with the software, but some experimentation and some common sense were put into use to see how compatible it really is.

The first point to note is the fact that 'EDIT line, increment' fails with an 'invalid parameter' error message, as does 'AUTO line, increment'. However, 'EDIT line' and 'AUTO line' both invoke a new screen editor very similar, actually identical, to Toolkit 2's ED command.

Other differences are outlined briefly in a 5 page Quill document that is supplied on the disc in '_doc' and '_lis' formats. The '_lis' file can be copied to any Epson compatible printer. (The disc is in MS-DOS format, not QDOS but the QXL can read both anyway.) The differences are :

- Multiple SBASIC jobs can be run concurrently. Each can be given a unique name if required.
- SBASIC programs can be EXEC'd even if not compiled. Parameters can be supplied if required.
- Resident extensions can be loaded so that they are available to every SBASIC job, or just one.
- A new variable, 'CMD$', allows programs to access their command line.
- VER$ can have parameters and returna 'HBA', '2.16', 163840 or 0. I assume that these extras are the SMSQ version and the system variables. The zero remains a mystery !
- Multiple in-line IF clauses work correctly even if not explicitly terminated.
- SELECT can use integer and floating point variables. Integer is faster.
- Integer FOR loops are allowed and are faster.
- Nested, in-line loops (REPEAT etc) work correctly, even if unterminated.
- REPEAT loops need not have a name.
- EXIT, NEXT and END something, need not have a name following. The correct program structure will be worked out.
- Multiple array/string indexes are not permitted, eg ARRAY$(1,2)(4), but can be easily by, eg ARRAY$(1,2,4) etc.
- String expressions can now be indexed, eg A$ = ('12345'& b$)(3 to 5).
- Sliced strings default to a start of 1 and an end of LEN(string). Zero length slices are allowed to be taken from the start or end.

- In order to make finding the different SBASIC jobs easier, Toolkit 2's commands and WTV take 1, 2 or 3 parameters to specify the mode and a pixel offset for the 0,0 pixel of the windows.
- QUIT and JOB_NAME are provides to make exiting from, and naming SBASIC jobs easier.
- Many SuperBasic bugs are fixed.
- SBASIC is fast !!!!!

To my mind, the nameless loops and the ability to EXEC SBASIC files are the best new features. One problem in writing procedures and functions in SuperBasic is the fact that on JS machines, JSU in the USA, using a total of 9 or more parameters and local variables in a procedure or function causes corruption of the program file when the program was run. This is fixed by QLiberator and Turbo compilers, but awkward when testing programs before they were compiled.

SBASIC has reduced the need for local variables used as REPEAT loops by allowing you to simply :

```
1000 REPEAT
1010  do_something
1020 END REPEAT
```

or alternatively,

```
1000 REPEAT
1010  IF condition THEN  EXIT: END IF
1020  do_something_else
1030 END  REPEAT
```

On the other hand, SBASIC does not appear to have the limit on parameters and locals.

When first loaded, SBASIC boots up in Monitor format and MODE 4, there is no pressing of F1 or F2 to select your display mode. Toolkit 2, which I think every QL user should buy, is included in the SMSQ file, so the first command will normally be TK2_EXT followed by WTV 4 to make life easier when editing SBASIC programs.

As outlined above, there are some differences between SBASIC and SuperBasic and until the compilers get upgraded, Turbo and Qliberator are going to have a hard time compiling SBASIC programs using these new facilities.

Both complain about unnamed REPEATs, EXITs and END REPEATs so if you need to compile your programs, they will require names. 'CMD$' will probably interfere with the Turbo Toolkit command with the same name and function. In testing, it appears to be ignored in a Turbo compiled program but works correctly when Qliberated.

To EXEC an SBASIC program, all you have to do is save it with the file extension '_bas'. Once this has been done, simply

```
EXEC filename_bas
```

alternatively,

```
EXEC filename_bas, "parameter", "parameter", etc
```

This will start up a new job in the QXL similar to EXECing a compiled task on a normal QL.

When an SBASIC program is EXEC'd, any filenames or devices supplied as parameters will be opened prior to the job starting and passed to the program as #0, #1 etc. To use Tony Tebby's example.

```
100 JOB _NAME 'UC'
110 REPEAT
120  IF EOF(#0): QUIT
130  BGET #0,a%
140  SELect ON a%
```

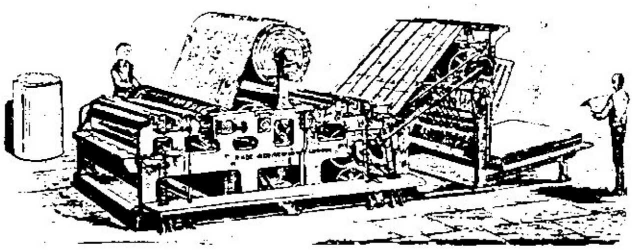
SBASIC - (CONT'D)

```
150      = 97 TO 122: BPUT #1, a% ^^ 32
160      = REMAINDER: BPUT #1, a%
170   END SELect
180 END REPEAT
```

Assuming that this is saved as 'UC_BAS', then 'EXEC UC_BAS, "file1", "file2"' will result in file1 being opened as channel #0, file2 will be opened as channel #1 and the contents of file1 converted to upper case and written to file2. If, while this is running, the Toolkit 2 command 'JOBS' is typed in, the jobs list will show a job called 'UC'. As a measure of the speed of this program, an 86 Kilobyte file of text, mostly in lower case, was converted in 38 seconds. Which is roughly 2,318 bytes per second.

An SBASIC program can also be used to 'filter' output from any program that writes to 'standard output'. This means channel #1 and I assume also C68 programs that use 'stdout'. To do this, simply 'EXEC myprog TO UC_BAS, "file2" and now everything that 'myprog' writes out will be passed in to 'UC_BAS' as channel #0 and be written to 'file2' in upper case. Very handy.

Stop the



SBASIC version 2.25

No sooner do I get my article sent off to Bob Dyl, than I get a letter and a disc from Tony Tebby in France with a new version of SMSQ/SBASIC which has corrected the bugs I was about to report and added some (many) new features as well. The following is a brief look at some of the new features.

QSAVED FILE

The reported problem with SBASIC being unable to QLOAD a QSAVEd file is all to do with the fact that QLOAD searches the QL ROM to find a bit of SuperBasic code. On the QXL under SBASIC this code is not there and so QLOAD should have done a tidy exit. It did not. What happened is that QLOAD followed the 'tidy exit' code and ended up somewhere in limbo and the QXL hung up.

Tony has fixed this problem by adding code to the SBASIC interpreter to allow the commands QLOAD, QSAVE, QLRUN etc to be used directly without the need to load the QLOAD_BIN file. In addition, there is also QMERGE, QMRUN and QSAVE_O.

This is not all. Tony has arranged things so that if you can LOAD, LRUN, MERGE or MRUN a QSAVED file as well - when you type in :

LOAD 'fred'

SBASIC tries to find a file called 'fred', if this fails, it looks for 'fred_bas' and if this fails, it will try 'fred_sav' before giving up if the file is still not found.

The 'Qxxxx' versions of these commands only check for the filename with a '_SAV' extension.

Just to keep things together, SBASIC can EXEC QSAVED files as well as _BAS files and normal EXECable programs - its getting so that compilers may well be obsolete !

And this is still not all. SAVE, LOAD etc all use default names as well. When you load a file you need to specify a filename, as above, various extensions are added to the filename in order to try to load it. However, when you come to SAVE the file, if there is no filename supplied to the QSAVE, SAVE, SAVE_O etc. command, SBASIC will use the same name as the file most recently loaded.

SBASIC - (CONT'D)

When a file is saved and a filename is given, the file's version number is set to 1. If the file is saved without a filename, the version number is set to be 1 greater than the version of the file that was originally loaded - this avoids the file's version getting higher and higher on each save.

HEXADECIMAL AND BINARY NUMBERS

SBASIC can now handle hexadecimal and binary numbers directly. To assign a value to a variable, all you have to do is this :

```
100 Hex_Number = $ABCD
110 Bin_Number = %01010101
120 PRINT Hex_Number, Bin_Number
130 PRINT $F0F0, %11110111
```

and so on. See PEEK and POKE for other uses.

EPROMS

There is quite a lot of software available for the QL on ROM. None of this software is much good on the QXL as it has no ROM slot to plug in to. SBASIC solves this little problem by supplying a command called EPROM_LOAD. This useful command lets you load a ROM image into your QXL and from then on, the QXL behaves as if there was a ROM in the slot.

You can, also EPROM_LOAD more than 1 ROM image file. The very first EPROM_LOAD after startup will load the ROM's code to address $C000 or 49,152 which is the original QL's ROM slot address. Subsequent use of the command will load the ROM image at a different address in the QXL's memory. This can cause problems with ROMs which assume that they will always be loaded at the normal ROM address.

ERROR TRAPPING

WHEN ERROR now works in programs, but is suppressed if you are typing a direct command. This keeps SBASIC from jumping into the error trapping routines every time you make a typing error while entering commands.

When any program stops with an error, they are reported as LINE_NUMBER:STATEMENT
so that you know exactly where on the line the error occurred.

CHANNELS AND FILENAMES

The commands LBYTES, SBYTES, SEXEC and their derivatives can now use an SBASIC channel number instead of a filename. For example :

```
1000 OPEN_IN #3, 'fred_bin'
1010 Address = ALCHP(FLEN(#3))
1020 LBYTES #3, Address
1030 CLOSE #3
```

This is useful as there might be times when you wish to carry out some other processing on the file via its channel number prior to closing it. It reminds me of my Spectrum, which if my memory is still working, could do something similar.

PEEK AND POKE

PEEK and POKE now have string versions, PEEK$ and POKE$ but POKE can handle strings correctly. Even or odd addresses cause no problems as either can be POKEd or PEEK$d.

The normal PEEK and POKE commands have been extended to allow PEEKing and POKEing into the System and Basic variables areas using offsets as opposed to absolute addresses.

SBASIC - (CONT'D)

Things get a bit complicated now. To PEEK or POKE in the System Variables area, you must prefix the first parameter with an exclamation mark (!) as follows :

```
PRINT PEEK_L(!! $20)
```

The first exclamation mark tells SBASIC that the PEEK is required into the System Variables area, the second says that this is a direct reference (see below) and finally, the hexadecimal number is the offset into the System Variables that is to be used. In this case, $20 or 32 decimal, which is the SV_RAMT or top of RAM +1 address.

Indirect addressing is when you access a value that is some offset from a System or Basic pointer. In this case, the command looks like :

```
PRINT PEEK_L(! $68 !0)
```

In this case, we are looking in the System Variables area as signified by the first Exclamation mark, the number that follows is the required offset, here we are using SV_JBBAS, or the base of the jobs table. Each entry in the jobs table holds the starting address of a job in the system, so an offset of zero, as signified by the number after the second exclamation mark means the base address of the very first job in the system, or the 'mother of all' SBASIC jobs. If we wanted to look into the Basic Variables area, we would simply replace the exclamation marks with backslashes.

BPUT AND BGET

These Toolkit 2 commands can handle string parameters so that multiple bytes can be PUT or GOT to/from a channel. BGET will also allow the getting of bytes from a channel to be stored directly in a sub-string of a DIMensioned string array.

ATAN

The ATAN function can now take 2 parameters which return quadrant results. Anybody know what this means then ? My maths is too far back in the past !

LINE EDITING

When entering or editing program lines, or values required by INPUT statements, many new editing functions are available. The normal LEFT, RIGHT and DELETE functions still work. In addition to these, there is now the ability to use TAB to move left or right by eight characters. Moving and deleting WORDs is possible as is deleting or moving to the start and end of the line. This brings the Basic editing into line with that provided under the Window Manager (WMAN) in the Pointer Environment.

FOREIGN LANGUAGE SUPPORT

SBASIC now supports English, French and German languages and keyboard layouts. This is quite simple to change and can be an endless source of irritation when someone comes along and sets your keyboard to the French, AZERTY, layout.

Languages can be set using either the international dialling code, 33, 44 and 49 or the car registration letter, F, GB or D as the single parameter to the LANGUAGE command.

There are 4 commands that are used to set your international requirements, LANGUAGE, LANGUAGE$, SET_LANG and KBD_TABLE.

LANGUAGE and LANGUAGE$ are used simply to tell you what language is currently being used or to find the language that will be used for a given dialling code.

SBASIC - (CONT'D)

PRINT LANGUAGE(977) would return the language that will be used if this were a NEPALESE QXL card, it seems to be English (44).

SET_LANG 'D' sets the language to German. This now causes all error messages to be printed in German as opposed to English.

KBD_TABLE 'GB' sets the keyboard layout to 'normal' QWERTY, while KBD_TABLE 'F' or KBD_TABLE 33 sets the French AZERTY layout.

Language and keyboard layout can be set at any time. Private or custom keyboard layouts can also be loaded.

NUL AND PIPE DEVICES

SBASIC provides us with a NUL device and named PIPEs. The normal unnamed PIPEs are still supported. I have not yet used these so I cannot comment on them.

CONCLUSION

So there we have it, Tony has been busy and once again has come up with the goods. I am more and more impressed with the speed and facilities that he has provided in SBASIC and only have one minor problem with it.

When writing programs for commercial purposes, you don't really want everyone to get their hands on your lovingly crafted source code do you ?

As Tony has created SBASIC to operate at near compiled speeds (QLiberator compiled that is) he thinks that there is no need to compile your code. This is very true, why compile it when you get very little in the way of additional speed ? The problem arises when you need to compile it and then you have to forget about all the new 'goodies'. Neither QLiberator nor Turbo can compile the additional features. We can't blame Tony for this, however.

Is there any development going on with QLiberator and Turbo ? Will we get new versions especially written for the QXL or will Liberation and Digital Precision not bother due to a limited market ? Who knows ? One thing that might help is the fact that SBASIC is also available for other machines as well as the QXL, this could be useful as these users may have requirements for compilers as well leading to increased sales.

This is all in the future of course, but I think the future is looking quite good now that we have a proper operating system for the QXL. Now if it only came with the Pointer Environment built in as well. (I thought that it did, but when I tried it, if didn't want to know, shame really.)

I have not been able to find any bugs in version 2.26 but I have not really had the time to get fully to grips with it.

So What are you waiting for, go out and spend lots of money on a QXL card for your PC and give it delusions of grandeur, or at least make it think that it is a real computer at last!

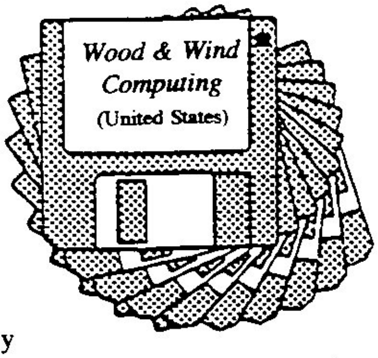
IQLR BACK ISSUES

Many of our new subscribers have asked concerning the availability of back issues covering our first three volumes. As we only maintain a limited supply of complete bound volumes, not individual issues, the demand has far exceeded our ability to supply.

In order to meet the needs of our subscribers, we are pleased to announce that Dilwyn Jones Computing has been granted the exclusive rights to supply QL users with back issues of IQLR. For information and pricing please contact Dilwyn (please note his adverts in this issue for address and telephone number).

An Accountant's View of QLerk

Grantham, Linncs., GREAT BRITAIN - John Taylor



To those who have had little or no training in accounting it should prove a boon as all the terms usually employed by the accounting fraternity are deliberately absent. Instead it relies on a good tutorial, common sense and a help facility.

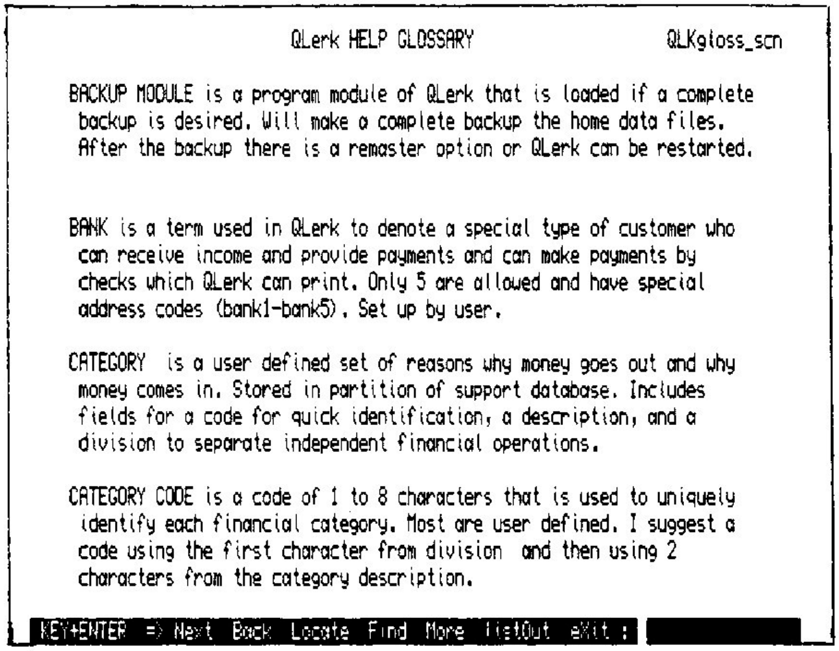
Essentially QLerk is an accounting database using the tried and tested Archive and as such it aims to record and provide whatever information you may require without the understanding of either databases or accounting. If you know anything about either then that is a bonus.Some may consider that Archive is too slow for a really large and complex taskslike maintaing the financial records of a small company but accounting is not an activity where speed is of the essence and the program has been developed to an extent that 2000 transactions between updates is adequate and all the essential information is retained on updating the system. The term 'Remastering' is used for 'Updating' but the text is such that understanding is always present.

If you make a mistake, and we all do, then the system allows you to make corrections, there is no limit to the corrections you can make and you don't have to make them immediately. If an entry is subsequently found to be incorrect then it can be amended provided the 'Remastering' has not been done in the meantime.

It is recommended that Trump Card with two 720k drives is the minimum hardware required with obvious advantages to those with Gold Cards, HD or ED drives etc.

The system is Menu driven using the cursor, the item number or the index letter in the Archive 'Screen' environment.

Such is the versatility of QLerk it is felt that only a very few would need all the facilities it has to offer. Never mind. You can use as little or as much as you want.



```
                    QLerk HELP GLOSSARY                      QLKgloss_scn

BACKUP MODULE is a program module of QLerk that is loaded if a complete
 backup is desired. Will make a complete backup the home data files.
 After the backup there is a remaster option or QLerk can be restarted.


BANK is a term used in QLerk to denote a special type of customer who
 can receive income and provide payments and can make payments by
 checks which QLerk can print. Only 5 are allowed and have special
 address codes (bank1-bank5). Set up by user.

CATEGORY  is a user defined set of reasons why money goes out and why
 money comes in. Stored in partition of support database. Includes
 fields for a code for quick identification, a description, and a
 division to separate independent financial operations.

CATEGORY CODE is a code of 1 to 8 characters that is used to uniquely
 identify each financial category. Most are user defined. I suggest a
 code using the first character from division  and then using 2
 characters from the category description.

KEY+ENTER  => Next  Back  Locate  Find  More  ListOut  eXit :
```

A Help System and Glossary is available from most Menus

As delivered QLerk will keep all the recognised financial records of Customers, Suppliers, Sales Income and Expenditure, Wages and Salaries, Overhead Expenses, and as many Bank Accounts as you wish to create. It not only keeps financial records but will also maintain 'Volume' records as well. There is a reporting system so that meaningful reports can be produced for differing purposes.

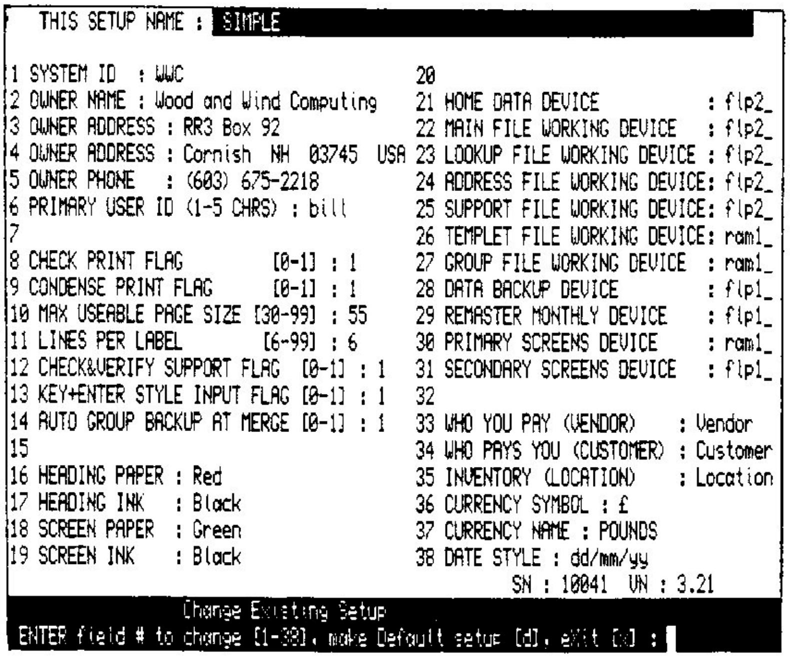
Though American in origin QLerk has been produced with the much wider world market in mind and much of this has been catered for when you enter the initial routines to create your'System'.

European users will have no difficulty altering such things as the currency etc and those familiar with Archive will have little difficulty with altering the spelling or even the menus. Most will probably decide to leave well alone anyway. However if you are going to print Checks, or Cheques, using the system then the presentation of the date will have to be altered. QLerk uses the American mm/dd/Year format and Europians will require the dd/mm/Year format.

Something which must be unique in accounting must be the help screens. I know of no other accounting program for any machine that will allow you access to such a facility whilst you are actually in the program.

An Accountant's View of QLerk - (CONT'D)

In addition, if you are so minded you can purchase a manual. It is not essential but if you wish to understand more fully what is happening or be tempted to adapt QLerk for a specific purpose then this is for you. There is no protection and the Archive progamming language is used throughout. There is over 150 A4 pages for a complete breakdown of the procedures though for setting up from scratch I found the Tutorial more than adequate. Just in case all that is not enough the author Bill Cable makes the offer to make custom modifications for a small fee.

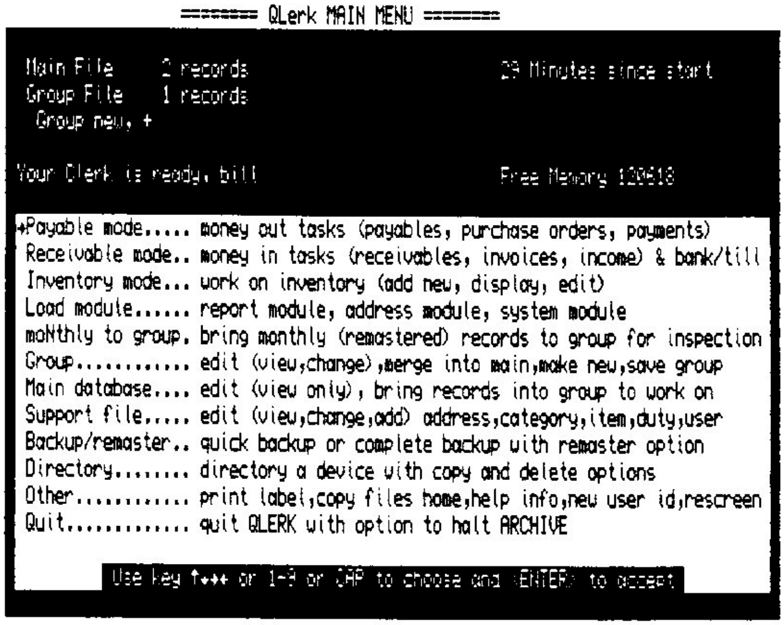


```
 THIS SETUP NAME : SIMPLE

1 SYSTEM ID  : WWC                         20
2 OWNER NAME : Wood and Wind Computing     21 HOME DATA DEVICE            : flp2_
3 OWNER ADDRESS : RR3 Box 92               22 MAIN FILE WORKING DEVICE    : flp2_
4 OWNER ADDRESS : Cornish  NH  03745  USA  23 LOOKUP FILE WORKING DEVICE : flp2_
5 OWNER PHONE   : (603) 675-2218           24 ADDRESS FILE WORKING DEVICE: flp2_
6 PRIMARY USER ID (1-5 CHRS) : bill        25 SUPPORT FILE WORKING DEVICE: flp2_
7                                          26 TEMPLET FILE WORKING DEVICE: ram1_
8 CHECK PRINT FLAG           [0-1] : 1     27 GROUP FILE WORKING DEVICE  : ram1_
9 CONDENSE PRINT FLAG        [0-1] : 1     28 DATA BACKUP DEVICE         : flp1_
10 MAX USEABLE PAGE SIZE [30-99] : 55      29 REMASTER MONTHLY DEVICE    : flp1_
11 LINES PER LABEL        [6-99] : 6       30 PRIMARY SCREENS DEVICE     : ram1_
12 CHECK&VERIFY SUPPORT FLAG  [0-1] : 1    31 SECONDARY SCREENS DEVICE   : flp1_
13 KEY+ENTER STYLE INPUT FLAG [0-1] : 1    32
14 AUTO GROUP BACKUP AT MERGE [0-1] : 1    33 WHO YOU PAY (VENDOR)       : Vendor
15                                         34 WHO PAYS YOU (CUSTOMER)  : Customer
16 HEADING PAPER : Red                     35 INVENTORY (LOCATION)     : Location
17 HEADING INK   : Black                   36 CURRENCY SYMBOL : £
18 SCREEN PAPER  : Green                   37 CURRENCY NAME : POUNDS
19 SCREEN INK    : Black                   38 DATE STYLE : dd/mm/yy
                                                   SN : 10041  VN : 3.21
                    Change Existing Setup
 ENTER field # to change [1-38], make Default setup [d], eXit [x] :
```

Setup easily modified at startup time to customize the system

It was noted that QLerk kept records of time taken etc., and though the clock is ever present it stood still until an action had been completed. eg. If you returned to the MAIN MENU then went to brew up, the clock still read 11:30 ten minutes later.

I have always felt that the QL is not over endowed with accounting packages though it was launched as a 'Business Machine' and the free software supplied of Abacus, Archive, Easel and Quill and these were all excellent aids to a small business. 'The QL Inteqrated Accounts' copies of which are very few and which is no longer offered for sale was a QL version of an early 'Sage Accounts' package supplied with the Amstrad PC. Add to that 'Cash Trader' and several 'Home Finance' programs and that was it.



```
                    ======= QLerk MAIN MENU =======

 Main File     2 records                      29 Minutes since start
 Group File    1 records
  Group new, +

Your Clerk is ready, bill                     Free Memory 120618

→Payable mode..... money out tasks (payables, purchase orders, payments)
 Receivable mode.. money in tasks (receivables, invoices, income) & bank/till
 Inventory mode... work on inventory (add new, display, edit)
 Load module...... report module, address module, system module
 moNthly to group. bring monthly (remastered) records to group for inspection
 Group............ edit (view,change),merge into main,make new,save group
 Main database.... edit (view only), bring records into group to work on
 Support file..... edit (view,change,add) address,category,item,duty,user
 Backup/remaster.. quick backup or complete backup with remaster option
 Directory........ directory a device with copy and delete options
 Other............ print label,copy files home,help info,new user id,rescreen
 Quit............. quit QLERK with option to halt ARCHIVE

          Use key ↑↓→← or 1-9 or CAP to choose and <ENTER> to accept
```

The MAIN MENU is the central branching point for all work

None of these were capable of producing a payroll, invoicing or printing cheques. Now all that is to be changed with the launch of 'QLERK'.

With all this sophistication you may consider that QLerk is going to cost you an arm and a leg. The cost of all this is all the more surprising considering that it took four years to develope the program to it's present level and all prices include postage etc. No need to worry, but I'll keep you in suspense, see WOOD & WIND COMPUTING's advert elsewhere in this issue for ALL the details.

RENEWING PRINTER RIBBONS

Massapequa, New York, USA - Bob Gilder

This article illustrates my trials and tribulations with some workable ideas pertaining to renewing printer ribbons, including some pitfalls. With the realization that the new breed of laser and ink jet printers are becoming more commonplace, I feel that there are a large majority of computer users who still use dot matrix printers. Perhaps, sometime in the future I will also go by the way of the newer breed of printers, and .... perhaps not!

I use a lot of printer ribbons during the course of a year and these ribbons tend fade after repeated use. Well back into the beginning of the 1980's printer ribbon costs were on the high side and that is what made me consider renewing and/or reinking my ribbons.

I contacted several ribbon manufacturers who advertised in Computer Shopper magazine and asked if they would sell me some ink - no sale! They were only interested in selling ribbon cartridges. Why miss out on a sale by selling ink to someone. However, one ribbon supplier mentioned that they use a quality ink containing a high quality lubricant for the purpose of lubricating the dot-matrix head pins... and that gave me an idea.

My Itoh Prowriter cartridge contained 18 feet of ribbon with a mobius loop. A mobius loop is a loop in which a one-half twist is made in one end of the ribbon before being attached to the other end. This assures that the ribbon turns over once every complete cycle for uniform wear and ink dispersion. Almost all ribbon cartridges have a pressed fit lid, which if you are careful, the lid can be pried off, exposing the packaged ribbon and gears.

First, you must remove the cartridge from the printer. To pry off a cartridge lid you must use a thin pen knife, or a thin blade common screwdriver or perhaps a spackling tool. Once the seam between the lid and cartridge has been exposed, carefully move the tool used for prying, around the entire cartridge until the lid has been removed.

I placed the opened cartridge on some newspaper on my garage floor and sprayed some WD-40 lubricant on to the ribbon and let it dry out for several hours before I tested it. The cartridge lid was replaced and then inserted into my printer for the 'acid test'. The worn out ribbon indeed had been rejuvenated, however, the result was far from being perfect. The WD-40 required more time to dry out; results indicated some small oily spotting on the printout. However, I was on the right track!

The cartridge lid was once again pryed off and I measured the area where the ribbon was stored within the cartridge. I drew a line across the inside of the lid and marked five places on that line so that I could drill holes to accommodate the small length of tubing which inserts into the lubricant spray head. Now I wouldn't have to pry off any more cartridge lids. This treatment for the Itoh cartridges lasted for quite a while before the lubricant 'fix' stopped working. The ink had to be replaced within the ribbon.

Again, browsing through pages of Computer Shopper magazine, I located a company selling printer reloads. Reloads are ribbons stored in a box, ready to be loaded into a used ribbon cartridge. I ordered three of these reloads and when they arrived, I attempted to 'byte the bullet', so to speak! The cartridge was opened; the original ribbon was removed (all 18 yards of ribbon). The idea was to place the box into the ribbon cavity, the fold down the front side of the box and carefully slide the contents of the box into the cartridge. What a job that was to accomplish and my fingers were full of ink (plastic dish washing gloves should have been recommended). It was now time to pull the ribbon outwards and run the ribbon through the ribbon guides inside the cartridge. This was accomplished in several minutes and then the cartridge was inserted into the printer. Actually, I believe this ribbon provided better ink characteristics than from the original cartridge.

Thinking about this project, I realized that the cost of the ribbon reload wasn't that much cheaper than purchasing a new ribbon cartridge. However, I still had two more reloads and both turned out to be a disaster when I attempted to install these reloads into the cartridges. As an old proverb states: "penny wise, dollar foolish".

Back again into the pages of Computer Shopper magazine for the next few months, carefully scanning pages for any printer ribbon company offering ribbon ink for sale.

RENEWING PRINTER RIBBONS - (CONT'D)

Eureka (an expression of triumph)! A company named Bede Tech, Inc., located in Cleveland, Ohio, offered quality printer ink in small quantities and a dedicated motorized ribbon reinker for the ITOH 8510A Prowriter (The Apple imagewriter). You guessed it, I immediately ordered the reinker and several additional bottles of ink. To this day, I still reink my ITOH ribbons with the 8 year old pint bottle of ink.

Now comes another question, "How do I know when the full length of the ribbon has been reinked?". The answer is this, place a small drop of liquid paper, (you know, it is the white liquid that you 'white out' mistakes with when using a typewriter) on the edge of the printer ribbon. When the ribbon has made the complete cycle, the white spot will show up and it is time to turn off the ribbon inker. Normally, I spot the edge of the ribbon on the top and bottom so that if the ribbon has a mobius loop (a twist) it will prompt the end of the reinking cycle.

As time went by, I purchased an old Epson MX80 printer at a swap meet at a reasonable price. You see, the ITOH printer wasn't Epson compatible and therefore, screen dumps from the ITOH were very small, whereas, the Epson printer screen dumps were almost full size on 8 1/2" x 11" printer paper. When it became time to reink the Epson ribbons, there again became a rude awakening - the ITOH cartridges were designed to rotate clockwise and the Epson cartridges rotated counter-clockwise and the pin which rotates the ribbon within the cartridges are shaped differently. So, here I go again!

By this time, cost of bulk ribbon purchases were dropping. Six MX-80 ribbons were priced at $1.55 each in lots of 6 from a company called MEI/Micro Center, Columbus, Ohio. (They also sold ribbons for the ITOH 8510-A at the same low price). The ribbons were pretty good for the price, however, it wasn't long before the MX-80 ribbons required a good drink of ink due to printing quite a few screen dumps. What to do?, what to do?

I decided that perhaps I could fashion a crank handle from a strip if maple, drill a hole in one end and then slit the very end to the hole. Drill a small hole horizontally across the slit and insert a 2-56" bolt through the hole. The larger hole would accommodate the plastic knob on the ribbon cartridge and the 2-56 bolt with a nut on the other side would be tightened against the cartridge knob. Another hole was drilled on the other end of the wooden strip. A 1" length of 1/4" dowel was drilled slightly oversized and a 4-40 x 1" bolt was inserted from the bottom of the wooden strip into the dowel and terminated with a loosely fitted nut.

The MX-80 cartridge was fastened on to my original reinker and the ribbon was then placed around the wicking roller saturated with ink, a drop of white out placed on the ribbon edges and then I started the cranking of the ribbon. Within approximately five minutes the ribbon was reinked. This type of reinking process will seem to go much more quickly if you turn on your TV set.

A fried and LIST club member, purchased a small bottle of printer ink and explored another method for ribbon reinking - he pulled out all of the ribbon from it's cartridge and applied ink to the exposed ribbon. The ribbon was saturated with ink, much too wet for use in his Seikosha MP-1300AI printer. He then hung the ribbon from a branch of a tree in his yard for several hours, where he then cranked the ribbon into the cartridge. At the time of this experiment, ribbons for his printer were selling for $17.00 each. It was well worth his efforts for trying out this method and he continued to do so for a couple of years until he finally purchased a ribbon reinker.

Along the way I realized that I needed a printer capable of producing letter quality output and decided to purchase a Panasonic KXP-1124, 24 pin printer. The printed material was beautiful, however, it really took a heavy toll on the ribbon. The ribbon is 4 yards in length and when the print became light, I had to place a ballpoint pen into a hole in back of the printer cartridge and push inwards. This process activated a spring steel leaf which placed pressure on the ribbon and a felt reinking wheel where it would rejuvenate the ribbon.

I purchased additional ribbons from MEI/Micro Center for $2.89 each in lots of six - a good price as far as ribbons for this printer was concerned. Original Panasonic ribbons were selling for $12.00 each! The MEI ribbons didn't hold up as well as the original Panasonic ribbon. So, it was off to the reinker again and as you may have guessed it by now, these ribbons rotated in a counter-clockwise direction and required a reinker with a tri-fin recepticle for rotating the ribbon. Never fear, I shall prevail! I produced a hand crank which fit the cartridge crank knob and proceeded to reink the ribbon. (what, again?). When the reinking process was completed, the cartridge covers were removed and I applied several drops of ink on the reinking wheel so as to extend the use of these cartridges.

RENEWING PRINTER RIBBONS - (CONT'D)

At the beginning of 1994, I finally purchased a new reinker from a company called V-Tech. This device is called V-Tech Universal Inker and provides two reinking outputs; counter-clockwise and clockwise rotations, and screw-in bits for matching any printer cartridge and universal clamps which act as a hold down for any printer cartridge. Diagrams for clamping ribbon cartridges are included for each bit ordered as well as complete instructions for using their reinker.

When I initially called V-Tech and advised them that I needed several driver bits to match Epson MX and Panasonic KXP-1124 printer cartridges, they recommended that I would not need a driver bit for the Panasonic cartridge and that they would send me a small reinking syringe for reinking the internal felt roller within the cartridge. I requested that they sell me the bit anyway ($5.00). V-Tech recommended that I drill a single hole in the top of the cartridge under the foam reinking gear and reink the gear with the syringe without removing the cartridge lid. (I removed the cartridge lid, drilled the hole in the lid and then replaced the lid onto the cartridge).

If you intend to reink your printer ribbons, please read on for some tips which will help you get the most from your reinking process.

1 - Inspect your ribbon for wear and tear. A worn ribbon can snag on the printer dot matrix pins causing pins to bend or fracture. This could result in a costly repair for your printer. Discard worn ribbons.

2 - Do place a spot of white out liquid paper on top and bottom of the ribbon before reinking so as to identify the end of a complete reinking cycle. You may want to time the complete reinking cycle so that you could use a timer on the reinker instead of using paint dots on the ribbon edge.

3 - Place a label on each of your ribbon cartridges and then write the date of reinking. After three or four reinking cycles, carefully examine the ribbon for any wear and tear.

4 - Use only quality ribbons for reinking, such as woven or seamless loops. Using ribbons with welded seams may force the splice to let go under the printer head, again causing damage to the printer head.

5 - Use only high quality ink, containing lubricated pigment free ink. Do not use stamp pad ink as they may contain water which can contaminate the printer head.

6 - When attempting to open a ribbon cartridge, do so carefully. Pry the lid a section at a time to prevent the plastic tabs from breaking and to prevent the internal gears from 'flying out' of the cartridge.

7 - When a reinked ribbon is installed in your printer, run off several pages of text which will bleed off any highly saturated areas of ink on the ribbon.

Note: If your user group meetings are light on attendance, perhaps a purchase of a club reinker could attract additional members to attend meetings. The LIST group has two members with Universal reinkers and we allow our members to reink ribbons either before or after our scheduled meeting. If a member requires a special 'bit' for his cartridge, he/she can purchase a bit for their own use. Usually two reinkings will pay for their purchase of a bit.

The V-Tech Universal Inker currently sells for $63.00 which includes a 2 oz bottle of ink which is enough to reink quite a number of ribbon cartridges. Additional quantities are available: 2 oz #2.90, 4 oz $4.80, 16 oz $15.75. They also carry inks for ink jet printers. They sell quality ribbon cartridges at low prices. Request a list of ribbon cartridges for sale. They will also provide ribbon reloads for your old ribbon cartridges. And they also sell remanufactured Laser Jet cartridges. Colored inks are available at a slightly higher prices. Colors available: Red, blue, green brown, yellow, gold or silver.

V-Tech can also supply inks for ink jet printers, 4-color heat transfer, indelible, UV resistant and OCR. Re-usable syringe and needle for ink jet refilling at $1.25 or 4 at $1.00 each.

V-Tech INC., 2223 Rebecca, Hatfield, PA 19440, USA
Telephone: 215-822-2989, Fax: 215-822-6394.

DIGITAL PRECISION MAKES ALL IQLR READERS AN OFFER OF THE FINEST QL SOFTWARE, WORTH £2100, FOR JUST £149.

DIGITAL PRECISION LTD

DO WE HAVE YOUR UNDIVIDED ATTENTION?

THE QL COLLECTION

	£
3D PRECISION COMPUTER AIDED 2D/3D DESIGN SYSTEM	49.95
ARCADIA GAME WITH BMX-BURNER & GRIDRACER	9.95
ARCHIVE TUTORIAL FOR BEGINNERS TO EXPERTS	19.95
BLOCKLANDS GAME WITH 65,535 4K x 8K SCREENS	9.95
COMPARE AUTO/MANUAL FILE MANAGEMENT UTILITY	19.95
CPORT SUPERBASIC TO C AUTO TRANSLATION SYSTEM	89.95
DAT-APPOINT APPOINTMENT ARCHIVE DATABASE SYSTEM	19.95
DISKTOOL WITH QUICKDISK DISK UTILITY	19.95
EDITOR SPECIAL EDITION ALL-FILE PROCESSOR	49.95
FONT ENLARGER ACCESSORY FOR PRO PUBLISHER	19.95
HARDBACK + FINDER HARD DISK BACKUP ENGINE	49.95
IDIS INTELLIGENT DISASSEMBLER SPECIAL EDITION	39.95
MAILMERGE DATABASE SYSTEM FOR ARCHIVE	19.95
MICROBRIDGE CONTRACT BRIDGE BIDDER & PLAYER	39.95
PAYROLL SYSTEM FOR UK SMALL COMPANY PAYE	49.95
PEDIT PRINTER DRIVER FOR PSION XCHANGE SUITE	19.95
PERFECT POINTER TOOLS FOR QRAM/QPTR USERS	29.95
PROFESSIONAL ASTRONOMER INTEGRATED SYSTEM	29.95
PROFESSIONAL PUBLISHER TOOLBOX PART ONE	29.95
PROFESSIONAL PUBLISHER TOOLBOX PART THREE	29.95
QKICK FRONT-END SYSTEM & DESKTOP MANAGER	24.95
QMATHS MATHEMATICAL SYSTEM PART ONE	69.95
QMON QJUMP MACHINE CODE SUPER MONITOR	39.95
RECOVER ENGINE FOR CORRUPTED ARCHIVE DATABASES	19.95
SEDIT + SCREENPRINT ARCHIVE DEVELOPER UTILITIES	19.95
SUPERFORTH FORTH-83 COMPLETE COMPILER SYSTEM	39.95
SUPER SPRITE GENERATOR GRAPHIC TOOLKIT	29.95
TRANSFER UTILITY SPECIAL EDITION WITH MODIFIER	29.95
ULTRAPRINT SCREEN DUMP ACCESSORY FOR EYE-Q	19.95
ZIP v2.0.1 PD FILE COMPRESSION / ARCHIVAL UTILITY	0.00

	£
ADVENTURE CREATION TOOL SPECIAL EDITION SYSTEM	49.95
ARCHIVE DEVELOPMENT SYSTEM + RUN-TIME MODULE	29.95
BETTER BASIC EXPERT SYSTEM AUTO PROGRAM IMPROVER	24.95
CASH TRADER v3.3 + ANALYSER (QL TRADING ACCOUNTS)	99.95
COPY UTILITY AND PRODUCTIVITY ASSISTANT	9.95
DATABASE ANALYSER FOR ALL ARCHIVE DATABASES	19.95
DIGITAL C SPECIAL EDITION COMPILER SYSTEM	49.95
DROIDZONE GAME WITH ACCELERATED GRAPHIC ENGINE	9.95
EYE-Q STATE OF THE ART QL GRAPHICS SYSTEM	39.95
GRAFIX QL 24-PIN UNIVERSAL PRINTER DRIVER	14.95
HEXDUMP UTILITY FOR ANALYSING MEMORY CONTENTS	9.95
LIGHTNING GOLD SPECIAL EDITION SUPER-ACCELERATOR	39.95
MEDIA MANAGER SPECIAL EDITION EMERGENCY/MANAGER KIT	49.95
NAMES AND ADDRESSES ARCHIVE DATABASE SYSTEM	19.95
PC CONQUEROR GOLD SPECIAL EDITION INCL. STD VERSION	99.95
PERFECTION PLUS SPECIAL EDITION W.P. WITH SPELLCHECKER	139.95
PROFESSIONAL ASTROLOGER PREDICTION/DELINEATION KIT	59.95
PROFESSIONAL PUBLISHER DESKTOP PUBLISHING SYSTEM	89.95
PROFESSIONAL PUBLISHER TOOLBOX PART TWO	29.95
QFLICK CARD INDEX QPTR DATABASE SYSTEM	29.95
QL DATA FILE COMPENDIUM FOR USE WITH DEMO PERFECTION	19.95
QMATHS MATHEMATICAL SYSTEM PART TWO & COMPENDIUM	59.95
QUICKLASER PUBLISHER DRIVER FOR HP LASERJETS & DESKJETS	19.95
REVERSI / OTHELLO GAME - WORLD CHAMPION STANDARD	9.95
SUPERBASIC MONITOR AUTO RUN-TIME TRACER/ANALYSER	24.95
SUPER BACKGAMMON GAME - CLUB PLAYER STANDARD	9.95
TOOLKIT III SUPPLEMENT TO SUPERTOOLKIT II COMMANDS	29.95
TURBO SUPERBASIC COMPILER v3.24 + TURBO TOOLKIT	99.95
XREF SUPERBASIC AUTO PROGRAM ANALYSER/REPORTER	29.95
UNZIP v5.0.1 PD FILE DECOMPRESSION / RETRIEVAL UTILITY	0.00

ALL THE ABOVE FOR £149

WHAT WILL THE QL COLLECTION COST ME?

Just **£149 in total**. There is nothing to add, no hidden VAT, and P&P to anywhere on earth is included. You save about £2,000. DP recognises that you probably have some titles already (though perhaps not the latest releases), and some may not be of interest to you yet (likely to change when you see them!) - the price more than fully reflects this. **THE QL COLLECTION** is worth it even if you only want to update all your existing DP products: however, DP will continue to accept orders for *individual* DP programs at the prices quoted above (+10% outside UK) if you insist!

WHAT EXACTLY IS INCLUDED IN THE QL COLLECTION?

You get the fullest, very latest, most up-to-date releases of all - **every one** - of the **60** QL programs listed above, *plus* a generous bonus of other, non-DP QL products. The software is DP's best, and thus the QL's very best. The titles would cost you over £2,100 (plus applicable P&P) to buy individually - you can check this by summing the above prices, or those quoted in earlier ads. The only titles omitted are Mega Dictionary (only for 2Mb RAM systems - add £20 for it), DR-DOS v7.0 / MS-DOS v6.2 (add £90) and any less capable variant of a title that is itself included in **THE QL COLLECTION** (e.g., since the top-of-the-range PROFESSIONAL PUBLISHER is included, DESKTOP PUBLISHER is obviously excluded). You get both versions of PC CONQUEROR (as the list shows) so as to cater for all hardware variations. And the very first order received also gets a QL modem/comms/network/ports package.

WHAT ABOUT THE PROGRAM DOCUMENTATION?

All, latest, applicable documentation (lots) is included on disk, and can be read and printed using Perfection Special Edition or Editor Special Edition, which are also both included, and which can of course be used to search, browse, analyse or edit manuals at your leisure. Printed copies may be bought later if wanted - details sent with order.

WHY CAN'T I FIND THE CATCH IN ALL OF THIS?

Because there isn't any. DP, whose QL commitment continues, makes this super offer to celebrate the birth of a marvellous baby daughter to Julie and Freddy. **THE QL COLLECTION** is licensed for use by the purchaser alone, who by buying it agrees not to resell or otherwise pass on any part of it, or of any DP software already possessed. Support is negotiable. DP *reserves the right* to withdraw **THE QL COLLECTION** offer at any time later than 14 days after publication of this issue of IQLR: orders for it received after withdrawal will be returned uncashed.

WHAT HARDWARE WILL I NEED TO RUN THINGS?

You will need a twin disk drive (DD, HD or ED, 3.5" or 5.25"), lots of blank disks and at least 1.5Mb RAM (Gold Card, Super Gold Card, QXL, ST/QL and equivalents) to fully use **all** the software. Many titles will, however, work on much smaller systems. Gold Cards ordered at the same time as **THE QL COLLECTION** cost only £120, complete with manual and 4-drive adapter, while stocks last (the original cost of this equipment being £390). Add £50 including P&P for an extra QL. Note that if no diskette size is specified when ordering, DP will assume 3.5" DD.

HOW AND WHEN CAN I GET MY COPY OF THE QL COLLECTION?

THE QL COLLECTION can be obtained - *immediately* - only by posting your order (including payment of £149 by cheque, postal order or cash, or quoting an ACCESS / MASTERCARD / VISA credit card number & expiry date) to:

DIGITAL PRECISION LTD, 222 THE AVENUE, LONDON E4 9SE

Town Crier Announcements of Upcoming Events

To have your Show, Workshop or AGM listed by the Town Crier, send all relevant information to IQLR's North American address. Please note deadline dates for submissions listed on page two of this issue.

12 November 1994 ***(SATURDAY)***

QL MEETING

St Joris College
Roostenlaan
Eindhoven
NETHERLANDS

20 November 1994 ***(SUNDAY)***

BRISTOL WORKSHOP

Contact: Mike Ashford
Tel: 0272 629981

Walton Park Hotel (10am - 5pm)
Clevedon
Bristol
GREAT BRITAIN

28 January 1995 ***(SATURDAY)***

QREVIEW QL SHOW

Contact: Bruce Nichols
Tel: 0708 755759

Rush Green Junior School (10am - 5pm)
Romford
Essex
GREAT BRITAIN

Masterpiece by Miracle Systems

The long awaited GRAPHICS CARD, now titled the "Masterpiece Enhanced Graphics Card" will soon see the light of day. The targeted selling price is £50 (may change) .

The small circuit board will replace the 8301chip (for those who have been replacing 8301's the cost of the board alone is well worth being done with the 8301)) and will cause you to remove the 68008 if you hadn't already done so.

There are two displays, QL mode (512 x 512) or an enhanced graphics mode (1024 x 512 - requiring an SVGA non-interlaced monitor). Software to drive the new board will reside in a new ROM to be issued for the Super Gold Card (required for enhanced graphics) Masterpiece will have 128K of video ram on board.

Those of you with Gold Cards will get the benefit of replacing the costly 8301 chip and the ability to upgrade in the future, who knows, maybe the Super Gold Card or maybe even the Super-Duper Gold Card (the last one, at present is only in our publishers mind).

What are you waiting for ?? Give yourself a Christmas present !! Call Miracle Systems for further information.

QUBIDE version 1.28

Troy, Michigan, USA - Don Walterman

QUBIDE is an IDE hard disk interface for the QL. Before describing it in detail, I'd like to explain how this interface ever came into existence. Months ago, one of the discussion threads on QBox was some wishful thinking about using IDE drives on the QL. IDE was discussed for some time when Zelijko Nastasic left some messages explaining that it should not be that hard to implement. In a short amount of time, Zelijko was asking for volunteers to test out his prototype IDE interface. The hardware was complete but someone still needed to write the software driver. Phil Borman then came forward and modified the Rebel driver to work with the new IDE card. Phil also wrote a set of programs to make life with QUBIDE easier. Add in Ron Dunnett for Finance and Marketing and you have a successful product. Others like Arvid Borretzen (who contributed Norback to the public domain) and Stuart Honeyball also played a part in making QUBIDE a reality. This hardly could have happened in the fiercely competitive PC market.

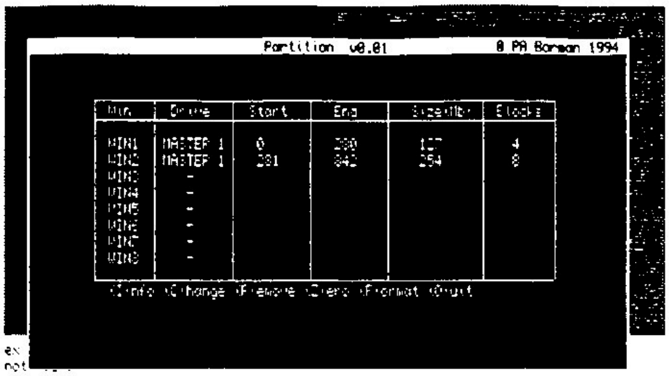
Hard disks on the QL have always been problematic until now. Miracle systems pioneered the Miracle Hard Disk. It worked but had its own unique requirements (plugged into rom port, one drive size) . The Rebel Disk was very promising but never made it into mass production. The Jurgen Falkenburg HDD card worked very well and sold in quantity. Its major drawbacks were the requirements for specific hard to find 8 bit MFM or RLL controllers and some compatibility problems with the Super Gold Card. Dirk Steinkopf released a public domain hard disk controller that worked but it required assembly from scratch as well as having most of the requirements of the JFC-HDD. Use of either the JFC or DS interfaces also required a bus expander to fit the interface adapter and the PC disk controller.

This brings us to the present. QUBIDE seems to be the ideal hard disk interface for the QL. The interface board is very small. The board plugs into the expansion connector and will fit completely inside the QL housing if it is the only board plugged in. It has a thru connector so any other card can be plugged in as well. QUBIDE uses readily available (inexpensive) IDE disk drives. Currently, I am aware of the following vendor drives in use with QUBIDE; Seagate, Western Digital, Conner and Maxtor. The only vendor that I would recommend avoiding at this time is Quantum. QUBIDE has been designed for future expansion. In the PC world, usually only 2 IDE drives are supported on a card. QUBIDE hardware will support up to 32 drives (16 Master-Slave pairs). The practical limit is somewhat less, but the board shows that it is designed for future developments.

I have had the good fortune to use a prototype QUBIDE for the last month or so and the production version of QUBIDE for the last week. Once QUBIDE was installed, I knew it would be a permanent part of my QL. This board works and works well. This review is based on my system consisting of a Super Gold Card 2.49, QUBIDE 1.28 and a Seagate ST3491A 420 MByte IDE disk drive.

Installing the drive is fairly simple. Plan on purchasing a standard IDE cable. The drives do not ship with one and neither does QUBIDE. Some drives do not have a connector for an access led. QUBIDE provides a connector for you. This is especially useful if you have your QL mounted in a standard PC case with the hard drive buried inside. I have my drive mounted in a standard 5 1/4" adapter bracket mounted in an open slot in a minitower case. The bracket has the led built in and it plugs directly into the Seagate drive.

The first thing you will notice is the new rom identifying itself as QUBIDE/Rebel 1.28.The following line will be the model number of the disk drive if it supports the proper IDE identify function. After your QL has booted up, formatting is nearly the same as a floppy. If your IDE drive is less than 128 MBytes or you plan to use only the first 128 MBytes, just enter format win1_"drive name". Accept the defaults and about 20 minutes later your drive will be ready to accept files. If your drive is larger than 128 MBytes, you have some options.



```
                    Partition  v0.01          © PA Borman 1994

     Win    Drive      Start    End     Size(Mb)  Blocks

     WIN1   MASTER 1   0        280     127       4
     WIN2   MASTER 1   281      842     254       8
     WIN3      -
     WIN4      -
     WIN5      -
     WIN6      -
     WIN7      -
     WIN8      -

     <I>nfo <C>hange <R>emove <Z>ero <F>ormat <Q>uit
```

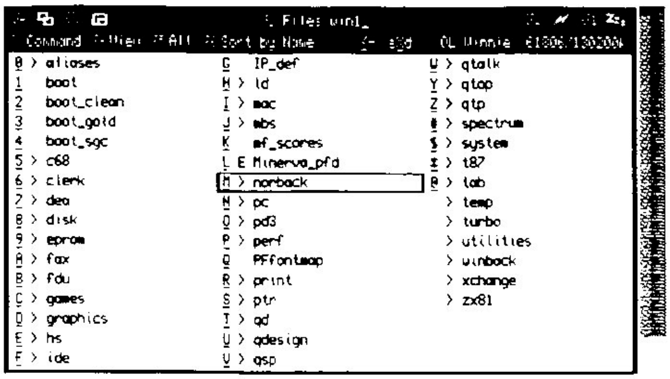
You can choose 8k blocks instead of the default 4k block size. This will make win1_ up to 256 MBytes. The tradeoff is some loss of efficiency. All files will now occupy 8k even if the file is only a few bytes long.

QUBIDE - (CONT'D)

You also have the option of specifying which portion of the drive to use for win1_. If you are not comfortable discussing heads, cylinders and sectors, you will want to accept the defaults. My drive currently is setup with win1_ as 128 MBytes of 4k blocks and win2_ as 256 MBytes of 8k blocks. That still leaves 36 MBytes to add in later as win3_ if I choose to. The reason you can't make your entire IDE drive respond as win1_ has to do with restrictions in QDOS related to a maximum of 65536 blocks per device. Back in 1983, I'm sure this seemed like more than the QL would ever need. With a microdrive only needing a maximum of 256 blocks, that left a lot of room for growth. Versions of MS-DOS prior to 4.0 could not handle disk drives larger than 32 MBytes in one partition. We shouldn't feel bad that QDOS from 1983 requires partitioning after 128 (or 256) MBytes. So, at this point your hard drive is formatted. If you want to partition it into win2_ , win3_ and etc. you need to use Phil Borman's partition program. Its quite easy to learn. I've included an example of my drive.

One point to consider when partitioning your drive is memory consumption. Each win you setup will require some system memory to maintain the directory in the QL memory. Setting up a number of drive partitions can use a sizeable amount of memory. That is part of the reason I settled on just win1_ and win2_. Carefully reread the section in the manual titled 'A word or two on partitions, block sizes and other concepts'. If you remember the days when floppy disk interfaces were coming out for the QL, we had the same situation.

Adding a floppy disk interface without adding a memory expander resulted in less available memory for programs. This is version 0.01 of the partitioning software so don't expect it to be perfect yet. The format routine within partition is not implemented yet. The other options do work. Change is used to add a new partition or modify an existing one. Zero allows you to mass erase everything on a partition. My system would not run either Phil's partition or WinEditor programs with the QDOS System-Extensions (Jörg Schiemann Software) loaded. Partition and WinEditor worked with all the other toolkits and extensions I use. WinEditor had some mysterious things happen like the 'tweed screen' reboot that I haven't managed to track down a cause for yet.



```
                          Files win1_
 Command   Wien   All   Sort by Name             QL Winnie
0 > aliases            G   IP_def              W > qtalk
1   boot               H > ld                  Y > qtop
2   boot_clean         I > mac                 Z > qtp
3   boot_gold          J > mbs                 # > spectrum
4   boot_sgc           K   mf_scores           $ > system
5 > c68                L E Minerva_pfd         % > t87
6 > clerk              M > norback             @ > tab
7 > dea                N > pc                    > temp
8 > disk               O > pd3                   > turbo
9 > eprom              P > perf                  > utilities
A > fax                Q   PFfontmap             > winback
B > fdu                R > print                 > xchange
C > games              S > ptr                   > zx81
D > graphics           T > qd
E > hs                 U > qdesign
F > ide                V > qsp
```

Now that you have your IDE drive formatted, what next? Why not take some time to plan how you'll use the space. With hundreds of megabytes available you will have to come to grips with subdirectories (if you haven't already). If you don't, your drive will rapidly have so many files in one directory that it will be difficult to find anything. I try to keep the root (top) level directory filled with nothing but subdirectories and the boot file. At the moment, my drive has 2619 files. Its very manageable with subdirectories. If you don't understand, ask for some help. This is a vital concept you need to master in order to be successful with a large drive.

I loaded software on my drive using the network to transfer all my files from my Gold Card-JFC HDD system to my new system. The network was the slow point here but it was much more convenient than backing up to floppies only to turn around and install from those floppies. This brings us to backups. My system backup runs about 19 ED disks. I don't back up my DOS partition from Conqueror or it would be 23 ED disks. This is getting to the point where a new backup device is necessary. I would think a tape drive would be the obvious choice. 120 and 250 Mbyte tape drives are available quite inexpensively now. Some use the high speed floppy controller that we already have on the Gold Card and Super Gold Card. Would it be possible to write a QDOS tape drive driver? Other possibilities are to buy a second IDE drive and use it to make an exact copy of your working IDE drive then store it in a safe place. Otherwise lets look at some numbers.

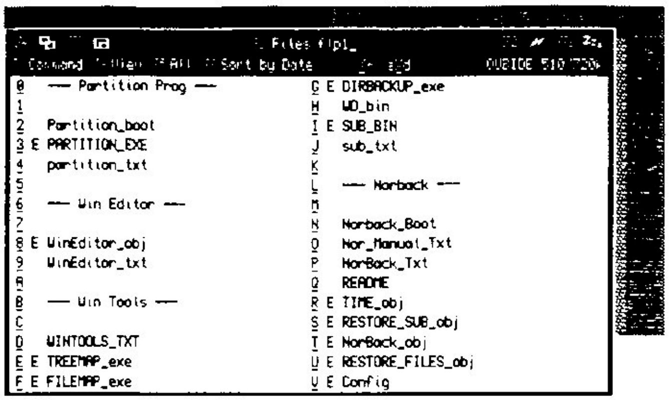
Media needed to back up a 420 MByte IDE drive:

1 420 MB IDE drive (used as a backup drive only)
or 2 - 250 MByte streaming tapes (if we had a QDOS driver)
or 132 ED disks
or 292 HD disks
or 584 DD disks
or 3819 microdrives

QUBIDE - (CONT'D)

The numbers are fairly staggering. Backups are mandatory. Since all hard drives will crash some day, you always need to be able to recover. The utilities disk supplied with QUBIDE contains Norback which is a very good backup program. Norback does require the Pointer Environment. The PE is not included with Norback since Norback is a Public Domain program while the PE is a commercial product. If you don't have the PE, Winback is a very good program that was reviewed in IQLR (Volume 3 Issue 5 page 10). Newer versions of Norback than the one supplied on the QUBIDE utilities disk are available from IQLR and most QBox BBSs.

The QUBIDE utilities disk also contains Phil Bormans sub device. It makes using sub-directories much easier. I prefer Phil's path device. I have a number of path_add statements in my boot file. That way I don't need to remember what directory a file is in. I can run acp (archivers control panel) with ex acp rather than ex win1_utilities_acp. The utilities disk also has some other maintenance programs to help you manage large numbers of files. The QUBIDE utilities disk provides a good complement of necessary programs to keep your IDE drive healthy. The QUBIDE manual does not detail the programs but hopes you will read the text files included on the disk.



```
0   --- Partition Prog ---         G E DIRBACKUP_exe
1                                  H   WD_bin
2   Partition_boot                 I E SUB_BIN
3 E PARTITION_EXE                  J   sub_txt
4   partition_txt                  K
5                                  L   --- Norback ---
6   --- Win Editor ---             M
7                                  N   Norback_Boot
8 E WinEditor_obj                  O   Nor_Manual_Txt
9   WinEditor_txt                  P   NorBack_Txt
A                                  Q   README
B   --- Win Tools ---              R E TIME_obj
C                                  S E RESTORE_SUB_obj
D   WINTOOLS_TXT                   T E NorBack_obj
E E TREEMAP_exe                    U E RESTORE_FILES_obj
F E FILEMAP_exe                    V E Config
```

Nothing has been said yet about how well QUBIDE works. I guess that is because it works so well you hardly notice it. It is fast. The new IDE drives are in the 12-15 msec access time range. Combine that with QDOS's efficient caching and you have a winning combination. One small bug in QPAC2 became apparent though. Any device with more than 100000k available may not display the drive name and file space properly in the upper left corner of the QPAC2 files display. This is only a problem with the default window size. If you have a number of files to display the window display is larger and the drive name and file space are displayed correctly. Phil is aware of the problem but can't address it since it is a QPAC2 bug. The Super Gold Card combined with QUBIDE shows how powerful the QL really is. I am a very satisfied user. Buy QUBIDE. You won't be disappointed.

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
☎ and Fax: 0203-501274 • Mailbox: 0203-591706

Lots of good news: first of all, Config Level 2 has been introduced. Programmers can get detailed information, Macros etc. by sending a formatted disk and return postage (2 international reply coupons). The advantages: once you configured a version of a program, say QD, the next time you get a new version of the same program you don't have to configure it manually anymore, MenuConfig puts the settings of the old QD automatically into the new version. There are more features, but, as upgrade prices are held extremely low, why not upgrade your software and see it yourself?

QD V7 exists - the main features are: Config Level 2 upgrade (of course!), an improved help system and Thing interface, the GOTO label, function or procedure list can be sorted alphabetically now, better word move with configurable word delimiters (to act exactly like SMSQ's word movement), moveable pulldown menus etc.

QSUP has been upgraded to Config Level 2. Minor improvements in the programs itself, DiskName is extended for HD and ED (and tested, as HD now works on the ST's under SMSQ). With new Thing Extension V3.

EPROM Manager has been upgraded to Config Level 2. It also contains some new features. With new Thing Extension V3, which will work better in the SMSQ SBASIC environment (open/use #0 if other channel not open etc.).

QMAKE, QBASIC, FiFi and QMENU have been upgraded to Config Level 2. HyperHelp has been upgraded to Config Level 2, and it may be configured to keep the list of functions and procedures when it goes to sleep, saving a bit of time next time you wake it.

WINED is a brandnew program: it is a real pointer-driven sector editor for floppy or harddisk. Many features, including partition-relative editing (on the ATARIs). Can be file-orientated or disk-orientated. And it is cheap too.

The QDOS Reference Manual will soon become the QDOS/SMS Reference Manual. The next bunch of updates will contain information about SMSQ, so everyone who is interested in the additional features of SMSQ should book for upgrade registration now. Some sets of upgrade sheets from last December are still available at DM 9,-.

SMSQ/E

SMSQ/E is getting more and more features than originally planned. By now you have probably read quite a lot about it, and there are still many things to tell you about it, unfortunately it is not space here. If you need more info about **SMSQ/E**, look at the previous issue of IQLR - there are many details, and, of course, you are welcome to write for a free catalogue with more (current) details about **SMSQ/E**. Or, even better, get yourself a version of **SMSQ/E** for your system(s) - it is not expensive!

QL Hardware & Spares

FLP/RAM Level 2 device drivers for SuperQBoard	**DM 56,-**
FLP/RAM Level 2 device drivers for TrumpCard	**DM 56,-**
SER Mouse software driver	**DM 40,-**
SER Mouse Package (mouse, adaptor & driver)	**DM 87,-**
ZX8301	**DM 19,90**
ZX8302	**DM 17,90**
QL Keyboard membrane DM 28,- 2 membranes	**DM 50,-**

QL-Emulators for ATARI

QVME - High-Res QL-Emulator for ATARI Mega STE and TT (up to 1024x800 pixels and more. PRICE CUT!! DM 499,-

EXTENDED4-QL-Emulator for ATARI 260ST, 520ST/STF/STFM, 1040ST, Mega ST (but not STE!!!) DM 289,-

Applications

QD	New Version	**[V7.06]**	**DM 125,-**
QBASIC	New Version	**[V3.00]**	**DM 49,90**
QDOS/SMS Reference Manual			**DM 84,90**
4 * Update sheets for Ref.Manual (incl. p&p)			**DM 30,50**
FiFi	New Version	**[V3.00]**	**DM 49,90**
QPTR		**[V0.25]**	**DM 92,-**
QSUP	New Version	**[V3.00]**	**DM 79,90**
QLQ		**[V1.13]**	**DM 69,90**
EPROM Manager	New Version	**[V3.00]**	**DM 61,50**
EasyPTR Part 1 DM 89,- Part 2 DM 49,- Part 3			**DM 49,-**
DataDesign V3			**DM 149,-**
LineDesign Version 2		**[V2.06]**	**DM 249,-**
HyperHelp for SuperBASIC		**[V2.00]**	**DM 49,-**
CueShell			**DM 95,-**
DISA		**[V2]**	**DM 95,-**
QMAKE	New Version	**[V3.00]**	**DM 44,90**
QMENU	New Version	**[V6.00]**	**DM 39,90**
QSPREAD		**[V1.26]**	**DM169,-**
HyperHelp for BASIC			**DM 49,-**
QMON/JMON		**[V2.10]**	**DM 89,-**
typeset 93-ESC/P2 (Stylus driver for text87)			**DM 69,90**
QLiberator	Price Cut	**[V3.36]**	**DM 139,-**

You can get Demo Disks of LineDesign, DISA, CueSehll or QSpread for DM 1,- each.

Upgrades

HyperHelp Upgrade from V1	**DM 6,-**
FiFi Upgrade from V2	**DM 6,-**
FiFi Upgrade from V1	**DM 16,-**
QMAKE Upgrade from V2	**DM 6,-**
QBASIC Upgrade from V1	**DM 6,-**
QMenu Upgrade from any Version	**DM 16,-**
QSUP Upgrade from any Version	**DM 16,-**
EPROM Manager Upgrade from any Version	**DM 16,-**
QD Upgrade from V6	**DM 16,-**
LineDesign Upgr. from V1	**DM 129,-**
DISA V2 Upgrade from V1	**DM 35,-**
QMON/JMON Upgrade from V2.xx	**DM 16,-**
QMON/JMON Upgrade from QMON only	**DM 32,90**
QSpread Update	**DM 16,-**

Games

Diamonds	(Action)	**DM 35,90**
BrainSmasher	(Strategy)	**DM 45,90**
Arcanoid	(Action)	**DM 35,90**
Firebirds	(Action!)	**DM 35,90**
SuperGamesPack	(5 various games)	**DM 90,-**
QShang	(Strategy)	**DM 45,90**
MineField	(Strategy)	**DM 39,90**
The Oracle	(Strategy)	**DM 49,90**
BlackKnight	(Chess)	**DM 119,90**
DoubleBlock	(Tetris)	**DM 42,90**
Lonely Joker V2	(6 Card games)	**DM59,-**
Lonely Joker V2 Upgrade from V1		**DM29,-**

WINED - Disk and File Editor Brandnew Program!

The author of FiFi (probably the most useful disk utility) has produced another very helpful program: WINED is a pointer-driven sector or file editor for FLP, RAM and, of course, WIN. It can even display a full sector at a time if you have an extended display (QVME or QXL).It can work partition-relative on an ATARI with many partitions and it has more than the usual options (search case dependent/independent, hex, write and append sector to a file, load from file etc.). It combines all the disk editing tools you had in one, being even more flexible and pointer-driven with all the features (sleep, move etc.). **Special introductory price DM 49,90**

If you require more information about any product, then please write for a free catalogue!

TERMS OF PAYMENT

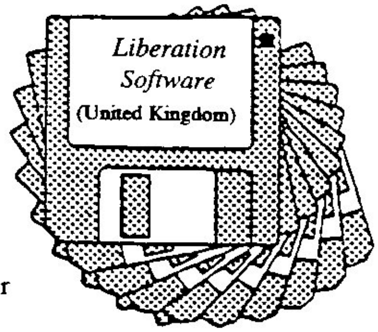
Postage and package (Europe) DM 14,- (if total value of goods is up to DM 50,- then only DM 9,-). (Overseas) between DM 14,- (1 item) and DM 35,- (maximum). All prices incl. 15% V.A.T. (can be deducted for orders from non-EEC-countries). E&OE. Cheques in DM, £'s, Eurocheques and Credit Cards accepted.

UPDATES

Updates of our software are usually free. The exception: major changes on a program (a new Version number before the '.'). Always send the master disk(s) to us, together with 4 international reply coupons for up to 5 discs or 8 IRC's for more. If you send updates together with a software order, then the return postage is covered by the wholesale postage. If a disk is faulty, add 1 IRC for a replacement. As the software changes from time to time, you may order a new manual together with the disc update. With upgrades, you automatically get a new manual.

Q Liberator A Review

London, ENGLAND - Mark Knight



Q-Liberator, in my view, is one of the three most important QL programs in the current scene, with Turbo and C68 making up the other two. Much of the commercial QL software available is written in SuperBASIC, and compiled with one of the two major SuperBASIC compilers. While probably the bulk of non-Pointer Environment (PE) software is compiled with Turbo, most of the PE software was created using Q-Liberator.

It could easily be argued that if SuperCharge, Turbo and Q-Liberator had never existed, then the QL software market might have died some time ago. Many commercial QL programs were written by hobby or part time programmers without the skills, or without the time, to tackle C or assembly language, and allowing them to produce fast, multitasking code using SuperBASIC has done much to keep the QL market going.

As a programmer, I am a keen, long-standing user of Turbo. I have experience programming IBM compatible PCs and DEC mainframes in PASCAL, Sinclair Spectrums in a peculiar mix of BASIC and fragments of Z80 assembly language, and the QL in SuperBASIC, with occasional forays into 68000 assembly language. I have come to regard Turbo as my favourite of all the compilers I have used, and so the opportunity to examine in detail a potential alternative was extremely welcome.

The above notwithstanding, I do not intend to write here from a position of prejudice. I hope I have examined Q-Liberator thoroughly enough, and impartially enough, for you to decide for yourself if it is a good choice for your own programming projects. Comparisons with Turbo are inevitable, but the two compilers are of such different type and philosophy that there is room for both in the marketplace. Some users will benefit more from one, some from the other, and I hope to provide enough information for you to choose for yourself.

First Impressions.

When I received the review copy, my first impressions were good. The manual seemed well written and the package simple to get started with. Preparing the Q-Liberator work disk was easy enough, and I installed Lightning SE, and oddly enough, the Turbo Toolkit. As many of the programs I have make use of the Turbo Toolkit keywords, I decided to see if Liberation software's boasts were correct. The advertising and documentation stated that Q-Liberator can cope with almost all QL toolkits, so why not try this one?

The other advantage of this was to make my own life easy. If Q-Liberator could cope with compiling Turbo Toolkit commands, this would leave me with plenty of SuperBASIC programs to do testing with, and no need to write or edit masses of files for this review. I did write and rewrite some code, but more on this later.

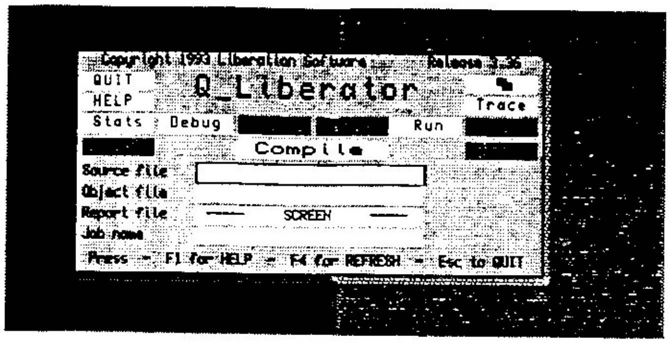
I actually found it very similar to Turbo to use. Typing "Liberate" at the command line replaces Turbo's "Charge" command, but once the compiler window appears there is a neat, simple to use front panel. This comprises a number of boxed options, and a box-cursor surrounding the current option. Move the box-cursor with the cursor keys, and press the spacebar to start compiling when "Compile" is highlighted. Some of the options require input when selected, and some toggle with the spacebar, and this again is just like Turbo.

For some reason, the similarity startled me, as I expected to find enough awkward differences to make this review hard work. The pleasant familiarity of the user interface meant I was quickly making use of Q-Liberator. The short programs I started with compiled in seconds, which was expected, and I had no trouble with them, which was also expected.

The dreaded benchmarks.

The first programs I compiled were a set of old Personal Computer World benchmark programs, to allow some comparisons with the interpreter, and also with Turbo. I do not hold great stock with benchmarks for many purposes, and I don't regard them as the ultimate test of a machine or compiler, but they can, if used with caution, provide some useful facts. In this particular case, comparing three versions of the same language, interpreter and two compilers, they do provide useful information.

Q_LIBERATOR - (CONT'D)



```
Copyright 1993 Liberation Software          Release 3.36
Source file
Object file
Report file   ——    SCREEN    ——
Job name
Press - F1 for HELP - F4 for REFRESH - Esc to QUIT
```

All tests were conducted a number of times, and the same methods were used every time to obtain average timings using the QL clock. To obtain fractional timings, tests were run up to 100 times and then the results divided by the number of runs to give what should be a pretty good figure. My system is a 16Mhz Gold Card QL, and I had all three modules of Lightning SE installed throughout testing. This represents my normal use of the QL, as I no longer use it for anything without Lightning installed. My QL has a JS ROM, which has little effect upon the results obtained, though it has some facilities which the AH and JM ROMs lack.

The first test was a modified text test, involving word-wrapping 1,000 very long words to a window on the screen, the window deliberately chosen so that only a few words would fit at one time. The small size of the window forced some scrolling to take place, and tested the speed of printing and passing parameters to the print routines. This was called TextTest.

Next, I used IntMath, a test of the integer arithmetic speed, RealMath for floating point calculations, and TrigLog, which uses a lot of SIN, COS,TAN and LOG function calls to test these floating point routines. GrafScrn is a poor test, but tests graphics plotting and I didn't have time to write a better one. The last test was Store, which opens a new file on disk, writes 1000 strings to it and then deletes the file.

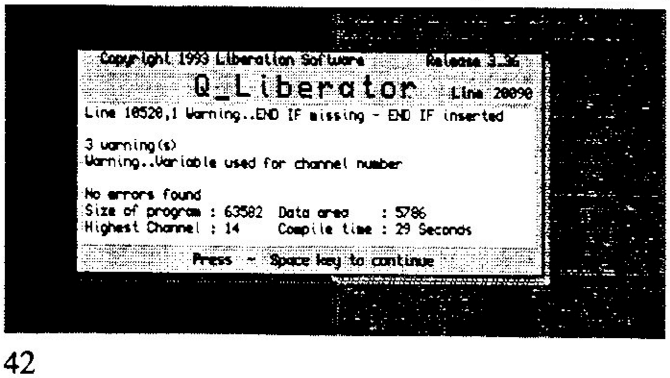
The table below summarises the results, all times are in seconds, and the speed index compares with an origianl IBM PC running GW-BASIC.

Times for:	Interpreter.	Q-Liberator.	Turbo.
TextTest	0.9	0.8	0.8
Store	2.6	1.4	1.2
GrafScrn	18.3	12.3	11.6
IntMath	1.7	0.1	0.08*
RealMath	1.4	0.4	0.2
TrigLog	4.2	4.0	2.1
Speed index:	8.74	21.03	28.26

* = This result more approximate than the others, and likely to be over.

These results were a little surprising to me, since although I expected Turbo to produce faster code I did not expect quite such a large difference. The first three results,TextTest, Store and GrafScrn, are of course "bottleneck" tests, in which the system is doing most of the work, and so the compiler has a smaller influence on the speed of the finished benchmark.

Benchmarks alone, however, might not represent a fair test, and so I decided to compile a SuperBASIC game that a friend has written, and compare the results using Turbo and Q-Liberator. This would be a more realistic test, as the program is larger than any benchmark I have seen, and has a wide variety of different routines in it. The important aspect of compiler performance is not how well it compiles benchmarks, but how well it compiles real programs.



```
Copyright 1993 Liberation Software       Release 3.36
              Q_Liberator         Line 20090
Line 10520,1 Warning..END IF missing - END IF inserted

3 warning(s)
Warning..Variable used for channel number

No errors found
Size of program : 63582  Data area    : 5786
Highest Channel : 14     Compile time : 29 Seconds
            Press - Space key to continue
```

Within minutes, I was able to load and compile the program, a share trading game with a few unique and even hilarious twists. A two-player game in which the computer played both players took about five minutes under the interpreter, but when compiled with Q-Liberator it took less than two minutes. I did not attempt precise timings, as these would be affected by the random events generated as the game runs.

Graphs, text and prompts appeared much faster, and the computer players were able to make moves almost instantly once Q-Liberator had compiled the game. The response of the program was dramatically improved, and of course there is the added bonus for the programmer that the program code is disguised by compilation.

As well as this, any compiled task could be multitasked under the Pointer Environment. As it isn't written to standard multitasking rules, this particular program will not multitask without special software, but Q-Liberator is perfectly capable of producing code that needs no extras to make it behave in a multitasking environment. As usual, this needs a little extra skill on the part of the programmer, and Liberation Software can't provide that.

Compiling the game had taken minutes, so it was a pain when I decided to compile the same program using Turbo. Turbo threw up a lot of error and warning messages, and the code needed some changes before it would run. Instead of the few minutes spent loading and compiling with Q-Liberator, I had to spend over two hours making changes to the code before Turbo would compile a working version. Part of this was because I had not written the code, and making changes to a program written by somebody else is always harder than changing one you have written yourself.

This reveals the main differences between the two compilers: Turbo is more fussy about the details of the code, Q-Liberator couldn't care less. Once again, when it finally ran the Turbo compiled version of the game was much faster, at least 40% faster overall, and in some places significantly more than twice as fast.

Turbo seems to optimise for speed much better than its rival, both for floating point calculations and in string handling. Heavy floating point calculation routines were about twice as fast compiled with Turbo, and the difference seemed greater where string handling was concerned, though I have not yet done definitive timing tests in this area.

I should stress that Turbo did not have difficulty compiling other programs I tried, they all compiled first time with both Q-Liberator and Turbo. The compilers themselves are both very easy to use, indeed I was struck quite forcibly by the similarity of the user interfaces that the programs use for their front panels.

Before leaving comparison with Turbo and concentrating on Q-Liberator, I decided to test and report upon two more bits of SuperBASIC. The first is a sort routine I have recently obtained, which (for those who know what this means) sorts in a time which is O(n) rather than the usual O(n*LN(n)). (Articles and SuperBASIC listing to be published soon, I hope, in both IQLR and QUANTA). This compiled easily with both Turbo and Q-Liberator.

The finished sort took 183 seconds to sort 5,000 items when running without compilation. Once compiled with Q-Liberator, it took 17 seconds, a superb and worthwhile speedup, I am sure you will agree. I did nothing to the code to take advantage of Q-Liberator's or Turbo's special features, by using integer variables for example. Turbo compiled the program to sort 5,000 items in 10 seconds. No problems with either version were experienced.

The last program to be compiled was an alarm clock program, written using many special features of Turbo, which I was mad enough to decide to rewrite just for this review. I decided to take on the job of replacing all of the specialised Turbo Toolkit code with Q-Liberator toolkit code or toolkit II code instead.

I didn't need to rewrite it this fully of course, since much of the Turbo Toolkit code can be happily compiled with Q-Liberator, but I really wanted to know how easy this kind of thing would be if you didn't have Turbo or its toolkit. This utility is used a lot by me to remind me when to stop programming (or writing reviews!) and go to bed.

I was pleased to find that it worked in the end, and without too much trouble. Although the pop-up menu responded a little more slowly, the extra delay was certainly not serious, and wouldn't have been noticeable if I hadn't been looking for it. The program became dependant upon the presence of Toolkit II, though this isn't likely to be a problem since almost every QL now has this on the disk interface ROM.

Q_LIBERATOR - (CONT'D)

The point of this conversion was to see how multitasking utilities of the kind I like to write fit into the Q-Liberator way of doing things. I had no trouble at all, and wouldn't anticipate that any competent programmer would run into difficulties with this kind of project.

Special features.

In order to take full advantage of Q-Liberator, it is best to read the manual thoroughly and use the compiler directives carefully. Q-Liberator is not fussy about your SuperBASIC, and will compile almost anything, but to get the best possible speed study and experiment is required, as with any compiler.

The simplest technique to grasp is that provided by the DEF_INTEGER directive, which needs to be placed at the start of your program, before any functional code. This instructs the compiler to treat variables, FOR indexes etc. as integers, which speeds things up enormously and also allows the finished code to be smaller as well. A line like...

```
100 DEF_INTEGER n,Counter
```

...would make "FOR n=1 to 10" and "FOR Counter=Start to Finish" both into integer FOR loops, and they would count much faster than a floating point FOR loop could manage. The machine code to handle integers is smaller as well as faster, so programs get smaller too, which in large programming projects is most welcome.

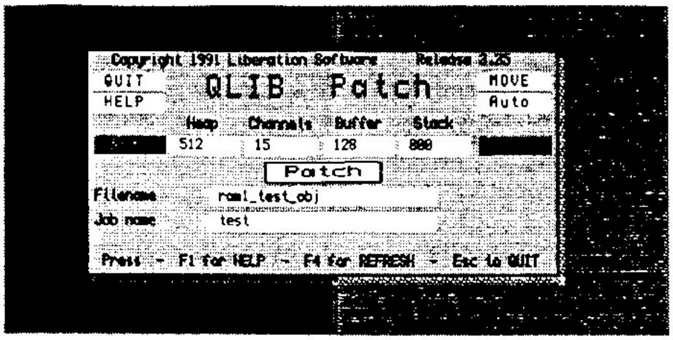
Q-Liberator also supports a number of other commands which affect compiled code only, embedded in REMark statements. One example is:

```
REMark $$i
```

This line in a program will instruct Q-Liberator to compile it in a form best suited to integer calculations. The other command in this set is:

```
REMark $$f
```

This, predictably, sets up the compiler to produce code better suited to floating point calculations. I was pleased to note that these can be used as many times as required in a program, to make different sections suited to the work required of them. Two of my favorite commands were to switch the compiler off and on again when compiling, so that debugging code can be used under the interpreter, but left out when the program is compiled. There are also good commands for setting dataspace memory requirements, stack size, heap size etc.



Copyright 1991 Liberation Software   Release 3.25

Heap	Channels	Buffer	Stack
512	15	128	800

Filename   ram1_test_obj

Job name   test

Press - F1 for HELP - F4 for REFRESH - Esc to QUIT

One excellent feature is the capability to instruct the compiler to include a ready-compiled section of code into the current program. This can save a lot of time in large programming projects, as modules can be written, compiled and debugged separately. A graphics or screen handling module could be written, and then simply included in any future programs without the need to recompile it each time.

This facility is not available with any other SuperBASIC compiler that I am aware of. Turbo users can achieve a similar effect, using link-loading of modules, but Q-Liberator's construction of a single executable file is much neater for most purposes. The ability is also included for two or more multitasking programs to share some or all of their FuNctions and PROCedures, which imitates the Turbo module link-load facility almost exactly.

Also unique in my experience of SuperBASIC compilers is the ability to use a form of program overlays. This allows a program to load a compiled module into RAM, use it, then if required unload it and load another overlay in its place. While sounding incredibly useful if you want to pack a big program into a small amount of RAM, I suspect that this facility will be little used.

Q_LIBERATOR - (CONT'D)

Most QL systems now have at least 640k of RAM, and the recent lower prices of Gold Cards, offering 2Mb and vastly increased speed, must have put paid to most worries about memory size. In addition, it increases the complexity of a programming job enormously to make efficient use of such techniques. The fact that the facility is available at all is a mark of the high level of sophistication of Q-Liberator.

An essential area of programming is trapping errors, in other words, what does your program do if the user selects the loading routine while there is no disk in the drive, or inputs a letter when the program needs a number? Q-Liberator implements an error trapping scheme of its own which allows a programmer to catch errors, though it does require careful use. I would expect an inexperienced programmer to struggle with it for a while, but it will reward study and experiment.

Also included is the facility to compile WHEN ERRor structures, found in bug-ridden form on QL systems with JS and MG ROMs, and also in a greatly enhanced form in the Minerva ROM. WHEN ERRor is a neat and conceptually simple system that would usually be easier to learn than the other system employed by Q-Liberator.

In order to use it, a section of code is set up like this:

```
WHEN ERRor
 IF ERNUM=-8  THEN RETRY 2040
 IF ERNUM=-20 THEN RETRY 2000
 RETRY 1000
END WHEN
```

In normal use, when the program reaches the WHEN ERRor it will jump straight to the END WHEN, just as it would if the structure was a PROCedure. However, if an error occurs it will, instead of stopping with a message, call the last WHEN ERRor routine passed over. This can use the ERNUM function to find out the error code, and RETRY or CONTINUE with a line number to restart the program at a line where the error can be dealt with. A simple WHEN ERRor routine like the one above is probably not adequate for most real programs, but it illustrates the idea.

WHEN ERRor routines can only be compiled on QL systems that have the facility on board (i.e. JS, MG or Minerva ROM), though once compiled the program will work in the same way on any QL. Users should note that WHEN ERRor is likely to cause severe problems, including system crashes, if used uncompiled on a QL without Minerva. Sometimes it will appear to work well with JS or MG systems, but not usually for long. If you want to use WHEN ERRor, you really do need Minerva, Q-Liberator or Turbo.

Documentation.

The manual was a refreshing experience. I found it clear, logically ordered and complete in its description of the compiler and its toolkit. This is unusual, and even though QL products often have more understandable manuals, this one stands out. I do have a couple of criticisms, one minor and one more serious.

Nitpicking, there are a number of editing errors left in the text, sentences with missing words, repeated phrasing etc. These don't actually present any real difficulty, but as it should be simple to root them out, and they are an irritating distraction, I thought it worth mentioning.

More seriously, I felt it would have been helpful if there was more discussion of how to write SuperBASIC code that takes best advantage of Q-Liberator's many features. It is true that Q-Liberator can compile almost anything that runs under the interpreter, but still, to make better use of the extra speed and new features a change of programming habits is certainly in order for some.

Sections on floating point optimisations, integer tricks, and also the kinds of expression and string handling that Q-Liberator can best compile would help. Advice on this sort of thing occupies over a hundred pages of the Turbo manual, and new users or those who don't understand are invited to skip it. Both of the two programmers involved in the writing of Q-Liberator are capable, knowledgable individuals, and passing on a little more of that knowledge would help some Q-Liberator users to make much better use of their purchase.

Q_LIBERATOR - (CONT'D)

These relatively minor quibbles don't detract much. The instructions and explanations are, as I have stated, clear, and I never found myself wondering what a feature was for, or how to incorporate it in my own programs. Sensibly, the manual starts with backup procedures and copying to a working disk, and progress to compiling your first program is rapid.

There is some advice, sound advice on writing multitasking programs, for one thing. The examples include an error trapping example showing how to prevent users from crashing your program by typing in text when you try to input numbers. There is a clear account of the error reports that Q-Liberator produces, both when compiling and when a compiled program crashes, which of course can still happen if the programmer has made a mistake (and even if I rarely make mistakes, you probably do it all the time).

The index is reasonable, perhaps a little sparse, though if I buy a compiler I normally read all of the manual anyway (at least twice) before I make any attempt to use it, so this wouldn't prevent me finding anything. The typography in this manual is ugly, with bad kerning in places, but as long as you can read it you probably won't care any more than I did.

Conclusion.

Overall, I was impressed by the quality of this product. Q-Liberator is a sophisticated tool for turning those slow SuperBASIC programs into much faster, multitasking machine code. You will still need to have some idea what you are doing to produce programs that behave when multitasking, but it will be a lot easier to learn the Q-Liberator way than learning assembly language. If you don't mind using EXEC_W rather than EXEC, then it is extremely easy.

I was a little disappointed by the floating point performance, but on reflection it is fast enough, just left a little in the shade by my old friend Turbo in this respect. I was able to use Q-Liberator within minutes of starting, and very few software products I have used would let me do that. In addition to the ease of use, there is more sophistication than I have space or time to describe. I won't be tempted to switch from Turbo, but I am not blind to Q-Liberator's attractions. Q-Liberator is simple to use, and has huge potential in the hands of a good programmer. It is also well suited to the relative beginner who wants to progress from writing SuperBASIC boot files to writing useful utilities. ***My verdict? Excellent.***

QUILL WITH AN ILL WILL

Oak Ridge, Tennessee, USA - Mel LaVerne

Recently, I stumbled across an anomalous (actually, berserk might be more like it !) behavior of Quill during the attempted editing of a large (106060 bytes) imported file. Memory should have been no problem; I was operating with Super Gold Card 2.49, with the assistance of Minerva 1.97.

Importing went smoothly enough, although lengthy (90 seconds).

"A Go To Bottom" gave 82 pages. Scrolling down showed the text to be present but with numerous forced page breaks. Since these forced breaks resulted in many very short pages, they were obvious candidates for deletion. I started by deleting the break at the bottom of page 2. This was when everything came unglued.

First of all, the entire text vanished ! Then I found multiple occurrences of a line from the index at the bottom of the file. This line appeared from line 11 of page 2 through line 35 of page 43. Furthermore, the maximum page number had become 49 ! There was now, of course, nothing left to edit.

I found the behavior to be quite repeatable, with the primary variation being in which line of the index was repeated.

Oddly enough, I was finally able to edit the file by resorting to Quill in the Xchange suite.

International QL News

This column is intended to bring together new and/or changing developments within the QDOS community. The information contained here has been reported by individuals and suppliers who desire to keep you informed as to their recent activities.

PROGS - Veltem, Belgium

The PROGS announce a 50% price reduction for PFlist and PFData, the new selling price is BEF 1000 each. In addition they have announced "THE PROforma FONTPACK" a collection of 100 high quality 'Bitstream fonts which can be used with all programs that use PROforma including LINEdesign.

LINEdesign is now to version 2.05 which includes PROforma version 1.04. This new version now contains a working Epson ESC/P2 printer driver and should no longer cause problems on Super Gold Card or Minerva systems. See our advert for additional information on our products.

JOCHEN MERZ SOFTWARE - Duisburg, Germany

Config Level 2 has recently been introduced, once you've configured a version of a program when you upgrade the software you won't have to configure it manually. MenuConfig will put your settings automatically into the new version.

With Jochen Merz precision, a new version of QD has been released (QD7) and the following programs have been updated with Config Level 2 : QSUP, QMAKE, FIFI, and QMENU.

In addition Jochen has released a brand new program titled "WINED", a pointer driven sector editor for floppy and hard disks. For more information see Jochen's advert elsewhere in this issue.

IQLR - Newport, Rhode Island, USA

As of this issue IQLR is a registered serial, the ISSN (International Standard Serial Numbering) number will now appear on every issue. In addition, IQLR is now available at the Library of Congress and is Internationally Copyrighted under the "Berne Convention Implementation Act of 1988. IQLR, International QL Report and ALL material printed therein is now protected by international law.

What does this all mean? Basically it means that IQLR is listed amongst most of the worlds publications. But, the most important fact is, that we are now protected against copyright infringment by international law, whereas in the past we were only covered by US law.

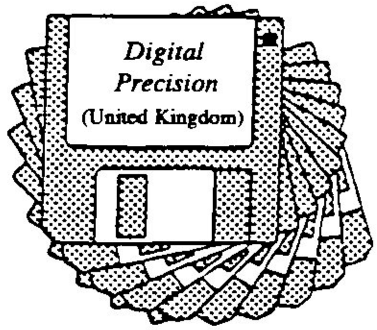
QUBBESoft P/D - Rayne, Braintree Essex, Great Britain

Ron Dunnett of QUBBESoft launched the long awaited IDE hard disk interface (QUBIDE) at the recent Scandinavian QL Meeting held in Gothenberg, Sweden. For more information please note the cover of this issue, the review of QUBIDE and QUBBESoft P/D's advert. (12 QUBIDE's already making their owners happy in the USA.)

With QUBIDE successfully launched, Ron would like to inform the readers of IQLR that he is building and supplying MINI-TOWER cases for your QL system. Give Ron a call for more information.

TURBO 3 (UNTOLD STORIES)

Oldbury, West Midlands, UK - Simon N Goodwin
(All Rights Reserved 1994)



Digital Precision's flagship SuperBASIC compiler has gone through three major revisions. Turbo was announced in 1986 and delivered the next spring, quickly followed by Version 2 with a revised manual, but version 3 was much longer in gestation, and has many hidden secrets.

This article explains the new features in Turbo Toolkit and the Turbo parser, notes bugs where they still exist, and discusses work in progress. I'm grateful to Chas Dillon for the new files, and some information about the changes; other details I have worked out for myself.

Qdos keeps developing, and many of the changes in Turbo over the years have been made to keep up with the latest fashion in ROMs and emulators. This article will obviously be of most use to those who own the Turbo compiler, in its latest or earlier incarnations, but it also contains many useful tips about Minerva, Amiga Qdos, Argos, QPAC and microdrives.

As usual I shall include some short snippets of program for you to type in. I'm grateful to Mark J Swift and Davide Santachiara, who developed the Turbo utilities presented here and gave permission for them to be freely distributed.

The latest version of Turbo is Parser 3.24, Codegen 2.15, Runtime Toolkit 3.20, and Turbo Toolkit 3.22. These have been provided as an upgrade for some time, but many new features are not explained. As the original designer of Turbo, and author of the majority of the code, I have experimented with the upgrade and compiled notes on the new features.

I should explain that I am writing as an independent QL enthusiast, and not as an employee of Digital Precision. I have had no dealings with DP since 1988, when I signed a contract ending my right to royalties unless sales pick up substantially, but I still take a keen interest in Turbo, which has been maintained by Chas Dillon.

Turbo 3 is the first version that will not run on a 128K QL, and it comes with Turbo 2 in case any users still have that requirement. The extra room has been used to add lots of new features, although few of these are mentioned in the manual.

I shall start by discussing the extra commands and functions in Turbo Toolkit 3. Many of these are accessible to SuperBASIC via the Turbo Runtime extensions (often called RUNTIME_EXTS or RUNTIME_CODE), so anyone with a copy of the runtime toolkit - supplied with many PD programs as well as commercial titles - can try them out.

New directives

Turbo Toolkit 3 includes a clutch of nice undocumented extensions, added by Chas for his own use, like a batch of directive extensions to configure the front panel controls. These are listed in the Table, with example parameters. The idea is that you include a set of directives at the start of each program, so that the appropriate front panel settings for the program are always used, regardless of your defaults. The parameter values are very flexible - you can use words, initial letters, or code-numbers from zero upwards.

Table - TURBO 3 front-panel directives

```
TURBO_objfil "flp1_Object_(output)_file_name_task"
TURBO_taskn "TaskNameTo12"
TURBO_repfile ""  (name; null indicates screen display)
TURBO_diags "INCLUDE" (line numbers) or O/I/D/DISPLAY/OMIT,0-2
TURBO_struct "STUCTURED" or S/F/1/0/FREEFORM
TURBO_model ">64K" or "> 64K", "<", "<", "0","1"
TURBO_locstr "IGNORE" or I/REPORT/CREATE/C/R,0,1,2
TURBO_optim "BRIEF" etc.
TURBO_wind 0 (number of windows to copy from Basic, 0-15).
TURBO_objdat 75 (a.k.a DATA_AREA)
```

TURBO3 - (CONT'D)

As you can see from the examples, the parameters allow most plausible mnemonics. They should be entered before the first executable line of the program, in the same area as GLOBAL, EXTERNAL and IMPLICIT directives. They are read before PASS 1, so they appear on the front panel, and over-ride the presets.

A couple of extra points are worth noting. If the task name specified in TURBO_taskn is more than 12 characters long it is truncated to that length. If you specify both TURBO_objdat and its forebear, DATA_AREA, the last to appear in the program takes precedence.

TURBO_P and **TURBO_F** are replacements for the PROCEDURE and FUNCTION functions, used with GLOBAL and EXTERNAL, as late versions of Minerva prohibit those names. You use these to indicate procedures and functions which are shared between separately compiled tasks.

MANIFEST assignment [: assignment]

This allows you to set manifest (=named) constants... but not expressions, at present, though you can use previous MANIFEST names. Chas intends that version 3.5, long mothballed, will fold constant expressions. This should fix the 22/7, 1/3 'integer expression' quirk in Supercharge and current Turbos!

DEFAULT_DEVICE **string$**

This is now equivalent to PROG_USE. Previously Turbo used its own 5 byte name buffer. Now it uses the Toolkit 2 one, so PARSER_TASK and CODEGEN_TASK can live in sub-directories. The buffer is created if TK2 is not installed. This change is most important for users with HD and ED drives, hard disks and networks.

IMPLICITL IMPLICITB IMPLICITC IMPLICITF IMPLICITD IMPLICITV

These new **IMPLICIT** types are extensions to Turbo's original set of three, and correspond to datatypes in the low-level C programming language. They are not used by Turbo but work as directives for Chas's SuperBasic-C-port. They allow you to specify C variable types to the translator while retaining SuperBASIC compatability on your QL or Qdos emulator.

DEBUG DEBUG_LEVEL

These are intended to provide conditional code sections, using directives to delimit test routines. I don't believe they work yet. Perhaps they are apocryphal, like THROW_AWAY, present but unimplemented since the first Turbo Toolkit.

Attempts to supply numeric parameters for DEBUG caused the v3.24 parser to fall over with spurious error reports; DEBUG "fred" was accepted but made no difference to the compiled task.

GetHEAD #channel% , address SetHEAD #channel% , address

These are extensions from the DIY Toolkit series, documented on DIY disk Volume H and in Sinclair QL World February 1988. Chas uses them to re-configure task dataspace without the need to load and re-save the entire file with LBYTES and SEXEC. This makes his configuration tasks for Editor and other programs much faster and more robust.

The next group of extensions deal with direct access to microdrives. There has always been a standard way to do this with floppy disks, but microdrives are another matter.

It helps to know the structure of a microdrive file system to understand these commands, but I do not intend to explain that here as it has often been done before, in Quanta and QL World among other publications. If all else fails you can learn a lot by calling GET_MSEC repeatedly and examining the results. This should be enough to recover program source or text from a cartridge with a corrupted directory or file map.

GMINFO address

You need a memory buffer to use this; you can allocate one with the function-call address=ALLOCATION(30). Before calling GMINFO, POKE the microdrive cartridge drive number in the first word, and zero in the second word, at the start of a 30 byte buffer at address. GMINFO reads the 10 byte medium name and two byte microdrive fingerprint number to the buffer several times over.

GET_MSEC address

This gets a microdrive sector into a 520 byte buffer. Put drive and sector number in the first two words at address, then a call to GET_MSEC will read the 512 byte sector thereafter, followed at offsets 516 and 517 by two byte values from the sector header; the block and file number respectively.

PUT_MSEC address

As for GET_MSEC, except you need to set the complete buffer before calling PUT_MSEC to write a sector. I must admit I have not been brave enough to try this!

These microdrive low-level sector routines were originally codged together by Colin Opie, and bolted onto Turbo Toolkit by Chas when he revamped Colin's Super Media Manager for DP.

I have not checked for changes since Colin's originals; Chas may have modified the code. It is most efficacious to read the sectors in the order they are recorded on the tape, not reverse order, as preferred by the original SMM! Of course Chas fixed this in his re-write.

SYS_VARS

This is the equivalent of the eponymous Thor function, and returns the address of the start of the system variables - normally, but not always, 163840.

Up to version 2.0, Turbo Toolkit expected the system variables to be at 163840, where Sinclair put them, but with the arrival of the Thor 20 the code was updated to call MT.INF when required, to find the system variables.

However two direct references remained in the Turbo code generator library, in the initialisation code that looks up Resident Procedure and Function addresses in the SuperBASIC name table. These are discussed in more detail later.

STRING%   STRINGF

These functions were added in Toolkit version 2.00, and do not figure in earlier documentation. They return the integer or floating point form of a string of two or six characters respectively - thus they are the complement of the FLOAT$ and INTEGER$ functions.

Unlike some similar extensions, STRINGF rejects invalid floating-point forms, with an exponent over 4095, which can crash the Qdos maths package.

Revision history

Turbo programs are made up of custom-generated code, data and library routines, extracted from CODEGEN_TASK and linked into the compiled task. The library was very extensively tested by Chas, Gerry Jackson and myself ('The Turbo Team') up until the release of version 1.14 in Summer 1987, and has not changed since. Changes have been concentrated in Turbo Toolkit and the translation program, PARSER_TASK, although CODEGEN_TASK has also been tweaked slightly in version 2.15 to cope with new inter-task parameter passing.

The first versions of Turbo Toolkit used an undocumented area at the end of the supervisor stack space to pass details between PARSER_TASK, Turbo Toolkit, CODEGEN_TASK and DATASPACE_TASK. This upset other people using the same space, so from Turbo Toolkit 2.10, Dataspace 2.2, Codegen 2.15 and Parser 2.04, the package communicates via an area in the Argos Objects list.

TURBO 3 - (CONT'D)

The Objects List is very similar in concept and execution to the Qjump Thing list, but based at $160(A6) not $B8(A6). Parser, Toolkit and code-generator all use this area so they must be upgraded together; you can't mix and match the files.

The current library version number is 5.10, and appears in bytes 3-6 of each compiled task. Version 5.09 was current until the release of parser 2.04. The only difference is the handling of DIM; version 5.10 includes improved code for the 'rubber array' feature which preserves data when an array is re-dimensioned.

The 5.10 library is prone to the effects of a minor ROM bug, which can also upset Supercharge. Negative parameters passed to RESPR in a compiled task crash the Thor and all Sinclair QL roms; most versions of Minerva report bad parameter if asked for a negative heap allocation. It's an error, so Minerva is right - it should be trapped.

Turbo Toolkit has seen a succession of small changes to improve compatability with Qdos variants such as Argos, QPAC and Minerva. Since version 2.00 the Turbo Toolkit calls MT.INF when it needs to locate the system variables, as is required for compatability with Thor 20/21 and Minerva's two screen mode.

The automatic cursor selection of CURSOR_ON caused problems for QRAM; I changed the code from Toolkit 1.39 onwards so it can be disabled by adding an exclamation mark after the channel number. It follows that CURSOR_ON #0! is preferred over CURSOR_ON #0 if you want your program to run happily under QRAM. You are also advised to provide a 'guardian window' that overlaps the entire display area used by your program, to make it easy for WMAN to swap screens.

Minerva fixes

Recent Minervae can't cope with Turbo's CHARGE, PROCEDURE and FUNCTION extensions, thanks to major hacks around version 1.7x; Laurence Reeves won't unbodge this, probably because his new tokenisation code is faster and the old code upset his MultiBASICs. PROCEDURE and FUNCTION are replaced with TURBO_P and TURBO_F respectively, but CHARGE remains a problem, not because of its name but how it works.

CHARGE loads the parser but gets the wrong parameter and fails to load the code generator, if used with later versions of Minerva. It's O.K. on early versions like Minerva 1.61, 1.63 and 1.64, but fails in 1.82 and 1.93, so you have to EXEC CODEGEN_TASK manually after successful parsing.

You can get around this by invoking the compiler with MERGE, as for SUPERCHARGE. This is what you need in the command file CHARGER; it should all be on one line:

```
EXECUTE_A PARSER_TASK; "RAM1_" : CLEAR : IF NOT PEEK_W(30 +
PEEK_L(SYS_VARS + 352)) : EXECUTE CODEGEN_TASK: END IF : END_CMD
```

Once you have put this in a file, with PRINT or an editor that allows long lines, you can invoke Turbo with:

```
MERGE FLP1_CHARGER
```

The command file checks the Turbo intercom area and the code generator runs automatically if the compilation is completed without error.

This example suits Turbo 2.04 and Turbo 3.24 with Minerva 1.93. "RAM1_" is the default task name, and can be adjusted to suit yourself. The code assumes that the Turbo intercom area is first in the Argos objects list at SYSVAR $160.L and the offset 30 to the parser error count is unchanged.

Minerva probably falls over because of unexpected adjustments to A6 as the CHARGE command works. These can also upset some of the Turbo string functions which (briefly) use an absolute pointer into Basic. The bug affects PEEK$ and BASIC_NAME$ in interpreted BASIC, but has no ill-effects in compiled tasks.

TURBO 3 - (CONT'D)

A few unintended limits are imposed by signed arithmetic, the bane of Psion and many 68000 programmers. Names more than 127 characters long prevent BASIC_INDEX% from searching the Name Table correctly. The correct, unsigned, limit is 255.

BASIC_NAME$ has two more imperfections; it can cope with no more than 32760 bytes of Name Table - over 4000 names - while the theoretical limit should be 64K, or 8192 names. You'll have to write a very, very big program to fall foul of this limit.

If you ask for the name of an expression entry, which has no name, BASIC_NAME$ does not report an error, as it should, but instead returns the first N% Name Table bytes, where N%=BASIC_B%(BASIC_L(24)-1). This is because the offset '-1' is stored where the function expects to find a positive name list offset.

The format of the configuration information in Turbo Toolkit changed for version 2.10 and 3.20. The configuration menu in UTILITY_TASK can be used to configure earlier versions, but later versions require you to use a SuperBASIC configuration program, TURBO_CONFIG_BAS.

Turbo checks the parameters of known extensions and reports errors, rather than simply passing the job on to the extension code. This makes compiled tasks more reliable, but stopped the compiler accepting new variants of existing commands, such as ATAN and VER$ in Minerva and SDATE TO on the Thor XVI. The compiler grammar has been relaxed to allow 'ad hoc' parameters in some such cases.

SDATE TO is now accepted, and CLOSE, WINDOW and PAUSE accept arbitrary parameters, but Minerva's VER$ and ATAN extensions still present a problem. The compiler library is flexible enough to allow multiple channel specifications in one PRINT or INPUT statement, as allowed on the Thor XVI, but so far the parser only recognises one #, at the start.

Minerva uses different program tokens from a 'real' QL ROM, although you can turn off this 'feature' - which accounts for the RENUM anomalies reported recently in IQLR - with the magic command POKE \\212,128. You need to do this before loading if you want to compile with Turbo or Supercharge. It's a pity Laurence didn't make compatability the default.

Two screens

Unless patched, most Turbo compiled tasks still do not run in Minerva's two screen mode. This is because of two direct references to system variables at 163856 in the ENTRY_ROUTINE used at the beginning of every compiled task.

In fact very simple programs (like PRINT "Hello World"!) that do not call resident procedures and functions will work in two screen mode without patches. Many QL commands are converted into Turbo code directly, with no need for the ROM code - but attempts to look up the names of any required extensions outside the task fail in two screen mode, giving a sequence of 'not loaded' messages after the task is loaded.

The relevant addresses can be patched to suit system variables at any other address, but then the task won't work with the variables anywhere else. Alternatively, the instructions can be patched where they appear in CODEGEN_TASK, so the changes affect all subsequently compiled tasks.

The disadvantage of this approach is that the patched tasks no longer work on a normal QL. This was not gggood enough for Davide Santachiara of Ergon Development, a very keen and prolific user of Turbo.

Davide wanted his programs to work without alteration on all systems, including in Minerva's two screen mode. The result is TURBOSTART, a short P.D. utility that modifies a compiled task, adding code at the start which checks the correct system variable base address and patches Turbo's 'entry routine' accordingly.

The result is shown in Listing 1 and Listing 2 - the return of the DIY Toolkit hex loader, which keen QL World readers will have typed in long ago! Listing 1 uses some functions from SuperToolkit 2, but there are equivalents in PD Toolkit (on the Amiga Qdos support disk) if you lack the commercial Toolkit 2.

TURBO 3 - (CONT'D)

(LISTING 1)

```
100 REMark TURBOSTART_BAS - Listing 1
110 CLS
120 INPUT 'Source file' !a$
130 INPUT 'Destination' !dest$
140 PRINT \'Converting....'
150 1p=FLEN (\'FLP1_TURBOSTART_bin')
160 1=FLEN (\a$)
170 d=FDAT (\a$)
180 IF  d<1024
190   Print a$! "is not a TURBO task!"
200 ELSE
210    ad=ALCHP (1p+1)
220    LBYTES FLP1_TURBOSTART_bin, ad
230    LBYTES a$, ad+1p
240    SEXEC dest$, ad, 1+1p, d
250    RECHP ad
260    PRINT dest$! "written."
270 END IF
```

First type in, SAVE and RUN Listing 2, which reads and checks the data. If all is well it creates a small code file, TurboStart_BIN, which Listing 1 combines with your Turbo task. The resultant task is only slightly longer and works regardless of the address of the screen and system variables!

(LISTING 2)

```
100 REMark Ex-Sinclair QL World HEX LOADER v 3d
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
130 CLS: RESTORE : READ space: start=RESPR (space)
140 PRINT "Loading Hex...": HEX_LOAD start
150 f$="flp1_TurboStart_bin": PRINT "Saving" !f$
160 SBYTES f$, start, byte : STOP
170 :
180 DEFine FuNction DECIMAL (x)
190 RETurn CODE (h$ (x) ) -48-7* (h$ (x)>"9")
200 END DEFine DECIMAL
210 :
220 DEFine PROCedure HEX_LOAD (start)
230 byte = 0 : checksum = 0
240 REPeat load_hex_digits
250   READ h$
260   IF h$="*" : EXIT load_hex_digits
270   IF LEN (h$) MOD 2
280      PRINT "Odd hex digit count in: ";h$ : STOP
300   END IF
310   FOR b = 1 TO LEN (h$) STEP 2
320          hb = DECIMAL (b) : 1b = DECIMAL (b+1)
330          IF hb<0 OR hb>15 OR 1b<0 OR 1b>15
340             PRINT "Bad hex digit in: " ;h$ : STOP
350          END IF
360          POKE start+byte, 16*hb+1b
370          checksum = checksum + 16*hb + 1b
380          byte = byte + 1
390   END FOR b
```

(LISTING 2 continued)

```
400 END REPeat load_hex_digits
410 READ check
420 IF check < > checksum
430   PRINT "Checksum incorrect. Recheck data. " : STOP
440 END IF
450 PRINT "Checksum correct, data entered at: "; start
460 END DEFine HEX_LOAD
470 :
580 REMark Space requirements for the machine code
590 DATA 88
600 :
610 DATA "6014000000004AFB",  "0000000000000000"
620 DATA "00000000000048E7",  "E09047FA003C7000"
630 DATA "4E41D1FC00000010",  "274800B027480114"
640 DATA "377C4E71006441FA",  "0020D1FC00000008"
650 DATA "47FAFFC6700636D8",  "51C8FFFC4CDF0907"
660 DATA "D9CEDDFC00000058",  "*", 7179
```

Re-located system variables may also cause problems for users of Thor 20, Thor 21, SMS-2 (depending on your configuration) and odd, early versions of Amiga Qdos. For best compatability emulator authors are advised to keep their system variables at 163840. Hopefully future system designers will find this convention easy to preserve, with hardware memory management available on 68030/40/60 chips.

Amiga Compatability

The Amiga Qdos emulator has developed into a very capable QL clone, particularly since the efforts of Mark Swift to improve compatability and disk speed, and the arrival of a cheap 68020 platform, the Amiga 1200. I run the latest version on a 68040-based Amiga 4000, and find it fast and reliable.

The Amiga's custom chips accurately emulate a QL MODE 4 screen by copying data from the 32K QL area to two 16K Amiga 'bitmaps'. This is very efficient, as the 'blitter' shares access with the processor, using alternate memory cycles to update the display, but it causes problems for one 68000 instruction, TAS, which is used in Turbo tasks, among others.

TAS stands for 'Test And Set' and expects to be able to read and write an address in two successive cycles. It fails on the Amiga, because the hardware gets in the way. The result is ugly but not fatal - you get incorrect display spacing in compiled tasks, because Turbo uses TAS to implement the "!" 'intelligent space' separator used in PRINT and INPUT.

The original Qdos emulator came with a 'TASfix' utility to replace TAS instructions with special F-line codes that the emulator could trap and perform as though they were real TAS instructions. This was fine, except that it stopped patched tasks working on the QL - Sinclair put code in the area Motorola reserved for the F-line vector!

Mark was not happy with this and came up with a program to patch out the TAS code with replacement instructions that work on all QL or Amiga systems. The result is TurboTASFIX_BAS, presented in listing 3.

(LISTING 3)

```
100 REMark -------------------------------------------------------------------
110 REMark TURBOFIX_bas by MARK J SWIFT plus tweaks by SNG
120 REMark Uses PDTK: ALCHP, CLCHP, FLEN, FTYP, FDAT, HEX$
130 REMark -------------------------------------------------------------------
140 :
150 DIM F$ (64) : OPEN #3; "Con_456x174a28x12" : OPEN #4; "Scr_104x12a362x20"
180 REPeat main_loop
```

(LISTING 3 continued)

```
190    WINDOW #3; 456, 174, 28, 12 : PAPER #3; 0 : INK #3; 7 : CLS #3 : BORDER #3; 3, 2
200    BORDER #3; 2,0 : BORDER #3; 1,2 : WINDOW #3; 438, 160, 36, 19
210    CSIZE #3; 2, 1 : PRINT #3; "TURBOFIX V1.01 (IQLR remix)" : CSIZE #3; 0, 0,
220    PRINT #3; "Program to replace a TAS instruction in TURBO compiled files."
230    PRINT #3; "The resultant code can be run from Amiga-QDOS or on a normal QL."
240    PRINT #3; \"NB: The original file will be overwritten, so back it up first!"
250    CLS #4 : BORDER #4; 1, 7 : INK #4; 4
260    INPUT #3; \"TURBO file to fix? >"; F$
270    IF F$= " " THEN EXIT main_loop
280    OPEN_IN #6; F$ : fl=FLEN (#6) : ft=FTYP (#6) : fd=FDAT (#6) : CLOSE #6
310    IF ft<>1 THEN PRINT #3; \F$; "is not a task!" : NEXT main_loop
320    base=ALCHP (Fl)
330    IF base>0 : LBYTES F$, base : ELSE PRINT #3; \"Out of memory!" : EXIT main_loop
340    X=FIND ("Turbo "&" Team", FILL$ (CHR$(223), 12), base, 0, 256)
350    IF X<>0
360     PRINT #3; \"First scan OK..." : CLS #4 : a=base : flg%=0
370     REPeat find_loop
380       BLOCK #4; ((a-base) /fl) *100, 10, 0, 0, 4 : p=PEEK_W (a)
390       IF (p=19128) OR (p=4050) OR (p=466)
400        RESTORE 910 : FOR i=0 TO 13 STEP 2
410           READ N : IF i<>0 THEN IF PEEK_W (a+i) <> N THEN i=0 : EXIT i
420        END FOR i : IF i
430           RESTORE 920 : FOR i=0 TO 13 STEP 2
440              READ N : POKE_W a+i, N : END FOR i
450           PRINT #3; "fixed at $"; HEX$ (a-base, 32) : flg%=-1
460    END IF : END IF
470    IF (p=19178) OR (p=4054) OR (p=470)
480     RESTORE 930 : FOR i=-10 TO 9 STEP 2
490        READ N : IF i
500            IF PEEK_W (a+i) <> N THEN i=0 : EXIT i : END IF
510          END IF : END FOR i
520        IF i
530         RESTORE 940 : FOR i=-10 TO 9 STEP 2
540            READ N : POKE_W a+i, N : END FOR i
550         PRINT #3; "fixed at $"; HEX$ (a-base, 32) : flg%=-1
560       END IF : END IF
570     a=a+2 : IF a> (base+fl) : EXIT find_loop
580   END REPeat find_loop
590   IF flg%<> 0
600      DELETE F$ : SEXEC F$, base, fl, fd : PRINT #3; \F$; "overwritten."
610    ELSE
620      PRINT #3; \ "No recognisable code - unknown TURBO version/fixed already."
630    END IF
640   ELSE
650     PRINT #3; \"Sorry - I cannot identify this as a TURBO file."
660   END IF
670   CLCHP : F$=INKEY$ (#3, 400)
680  END REPeat main_loop
690 CLOSE #3 : CLOSE #4 : STOP
700 :
710 DEFine FuNction FIND (txt$, msk$, base, s, e)
720 LOCal i, j, K, L
730 CLS #4 : L=1 : i=s
740 REPeat i_loop
```

TURBO 3 - (CONT'D)

(LISTING 3 continued)

```
750     j=0 : REPeat j_loop
770       K=0 : Repeat k_loop
790         IF (PEEK (base+i+j+K) && CODE (msk$ (K+1))) < > CODE (txt$ (K+1)) && CODE
               (msk$ (K+1))

800               EXIT k_loop : END IF : K=K+1
820             IF K=LEN (txt$) : L=i+j : EXIT i_loop
830             END REPeat k_loop
840              j=j+1 : IF j=50 : EXIT j_loop
850        END REPeat j_loop
860      IF i>=e : BLOCK #4, 100, 10, 0, 0, 4 : ELSE BLOCK #4; ((i-s)/(e-s)) *100, 10, 0, 0, 4
870      i=i+50 : IF (i-e)>=50 : EXIT i_loop
880     End REPeat i_loop : RETurn L
890 END DEFine FIND
900 :
910 DATA  19182, 143, 32256, 29184, 20112, 17393, -6144
920 DATA  2286, 7, 143, 32256, 29184, 20112, -11314
930 DATA  12842, 34, 8775, 19008, 26410, 19178, 23, 26404, 10249, 14849
940 DATA  14890, 34, 8775, 19008, 26410, 2282, 7, 23, 26402, 10249
```

It is worth using this even if you do not have an Amiga, because it means that your programs will be compatible with Amiga Qdos as well as all the other variants. Public Domain suppliers can use it to ensure that Turbo tasks in their collection are Amiga-friendly.

The Future

For my own amusement I have developed an improved version of Supercharge (version 1.21) which dispenses with the lenslok copy-protection code, present but disabled in the current 'Special Edition' version 1.19. The new parser is faster, leaves about 5K more memory for user programs and supports extra options to reduce code size.

If money becomes available to fund further development of Turbo, Chas hopes to implement constant folding, an IMPLICITV directive for ROMs that change the parameters of Sinclair extensions, and a buffer size directive - the only one missing from the Turbo 3.2 set.

I'm also trying to persuade him to permit direct access to the 'vector table' that points at routines, arrays and variables in compiled tasks, as this would facilitate direct array manipulation and function and procedure parameters.

Re-entrant code

In theory Turbo tasks could be made re-entrant, so you could run multiple instances with only one copy of the code in memory, shared by all the running tasks. Many changes would be needed in the library, and at the moment it is so reliable that I shrink from the idea of tweaking it. A lot of carefully planned testing would be needed.

Possible advantages are smaller tasks, through a QLib-style shared library of threaded code, execution directly from (EP)ROM (with no need to copy to RAM first), and shared use of ROM or RAM code by several tasks, e.g. concurrent editors.

Turbo threaded code contains a sequence of library routine addresses, expressed relative to a point 32K after the start of the library (to allow 64K of signed offsets). It would not be possible to share the library between two tasks compiled with different versions of the library, as offsets would differ. The thread word to execute any template depends on the amount of code earlier in the library.

Sharing the library would mean that CODEGEN would need a tweak to make it write tasks assuming a full library, while writing that library to a separate output file. TURBO was designed to generate concise, single tasks. This

TURBO 3 - (CONT'D)

stems from the design restrictions on Supercharge, which had to fit a sophisticated parser into 128K, with a substantial program alongside. TURBO has many more features and much faster string-handling, but the internal datastructures are much the same as in Supercharge.

As RAM gets ever cheaper, re-entrancy is less valuable (and complicated). Turbo Toolkit is re-entrant, anyway, so Turbo tasks and SuperBasic share some code already.

Overlays

LINK_LOAD, separate compilation, and the sharing of code and variables between tasks, are probably the most powerful features in Turbo. Many users seem to have trouble getting to grips with them, perhaps because they do not appear in most other compilers or languages.

Another planned but unimplemented feature is 'overlay code', in which one task can load and link another, use its routines for a while, and then release it to free memory, perhaps to load another overlay module. The LINK_LOAD code in Turbo has always been designed to support this, and it may appear in a future release.

The Final Frontier

My favorite proposal is NVECTOR, for direct access to the vector table. This works rather like 'VARPTR' in Microsoft Basic, but is much more fun in a multi-tasking 32 bit linear address space.

All it takes is a single function that converts a SuperBasic literal name into the corresponding 32 bit address vector. It returns the code or value offset from the Name Table under the interpreter - a bit different but it's the closest we can get.

This allows fast SAVE/LOAD/EDIT/SEARCH/SLICE operations on arrays of all types and sizes. A few useful demos would need to be written; INARRAY with SEARCH_MEMORY, SAVE_ARRAY, INSERT_SLICE via MOVE_MEMORY, C-style string operations, etc. They could be made to work under the interpreter, too. Descriptors are different, but equivalent.

Array Files would match the existing format of Turbo Virtual arrays, i.e. nothing but binary data; strings could be fixed-length, as in memory, or packed to suit the Turbo Toolkit function GET$, with 16 bit lengths and byte-aligned text.

The same function would let real enthusiasts block, divert or pass resident procedure or function code address vectors between LINK_LOADed modules running concurrently. It's risky, but fun. Try that in Ada!

XCHANGE BOOT

Massapequa, New York, USA - Bob Gilder

In correspondence with Ruth Fegley a member of the CATS group and IQLR subscriber, I stated that I would like to permanently change the default DATA drive from flp1_ to flp2_. Recently I received a note with a two line addition to the Xchange Boot program. Ruth stated that a member of CATS provided the following SuperBasic listing that really does the trick.

```
265 DATE_USE flp2
267 PROG_USE flp1
```

Note that there is not an underscore after the drive designators when the DATA_USE and PROG_USE PROcedures are defined.

THE SUPER GOLD CARD

Shelby Township, Michigan, USA - John J. Impellizzeri

OK the title is a little play on words. Miracle's new Super Gold Card is quick! Even if you already have the original Gold Card, the Super Gold Card improves on it in many ways including speed, cpu, memory and additional features.

Just what is the Super Gold Card? It is an expansion card for the Sinclair QL that gives you 4 megabytes of RAM, built-in Toolkit 2, a battery backed clock (that works), floppy disk interface for up to four DD, HD or ED disk drives, a Centronics parallel printer port and a 68020 microprocessor running at 24 MHz. There is an on board socket where an external 5 volt power supply can be connected if you have your QL modified in any way such as in a PC type case using its power supply. Of course, the original QL power supply can also be used with no problems to run both the QL and Super Gold Card. A Centronics printer cable is included as is an excellent manual that covers Toolkit 2, the original Gold Card and its features and the Super Gold Card and its extras. The Super Gold Card is covered by a 2 year warranty. See Miracle's advertisement elsewhere in this issue for current prices and trade in deals.

Looking at the Super Gold Card (SGC) shows the usual Miracle quality design and construction. The components on the board are densely packed but neatly arranged. When you stop to consider all the circuitry that is contained on that small piece of printed circuit board, it is a miracle! (Another play on words!!??!!). It is just a bit longer than the Gold Card but still leaves much less of itself sticking out of the QL than the Trump Card. About the only thing you see is the large heatsink (which does get quite hot), and the two floppy disk drive connectors. Underneath the heatsink, facing out to the left of the QL is the connector for the parallel port. Also underneath, facing to the back is the auxiliary 5 volt power supply connector. Full specifications for the connector and its use are in the SGC manual. While the heatsink does get quite hot during operation the SGC has been 100% reliable running for days at a time with no problems.

There are two connectors for interfacing to disk drives. These are the same type as used on the Gold Card, Trump Card and many other QL disk interfaces. The way that Miracle has them set up allows for a lot of flexibility in connecting drives. If you have 4 drives that can be configured as drives 1 thru 4, then they can all be on the same ribbon cable and plugged into one connector. If you have 4 drives, but they can only be setup as drive 1 or 2, then set up a pair as 1 and 2 and the other as 1 and 2 and plug each into the two connectors on the SGC. The drives will still be accessed as drives 1 thru 4. This arrangement also gives you the advantage of using drives set up as 1 and 2 on another system and moving them over to the SGC easily when you need extra drives temporarily, without having to open up a drive case and change tiny jumper plugs.

The drives to be connected can be double density (DD), high density (HD) or extra density (ED) giving 720K, 1.44M or 3.2Mbytes, respectively. You can use any mixture of drive types, they don't all have to be the same type. However, you do have to have an HD drive to get 1.44Mbytes or an ED drive to get 3.2Mbytes from the proper floppy disk. Note that having four ED drives online gives you 12.8 Mbytes of storage! Also note that these higher density drives can transfer (save/load) data faster than a standard drive when using HD or ED disks.

The parallel port is a very welcomed addition in my opinion. If you do any type of graphics printing and your printer has a parallel port, you will notice the difference. Even though I have a Deskjet which is capable of being driven by the serial ports at 19200 baud, there was a very noticeable increase in print speed using the parallel port to print graphics. Screen dumps really fly now! Straight text printing is probably technically faster but I didn't notice any change using the parallel port versus the serial port. However, any word processor or desktop publishing program that prints text using a graphics mode of the printer will be sped up considerably. The included cable means you can plug right in and start enjoying the benefits. Using the port is simple, there is a new device called 'PAR'. Just print to that rather than 'SER' in your application and you're all set! If you have an older or misbehaving application that insists on printing to a serial port, the SGC does give you commands to trick the program into thinking its printing to a serial port but in reality its output has been redirected to the parallel port.

So where does the SGC get its' speed? From a few things: First, the cpu, which on the original QL, is a Motorola 68008, internally a 32 bit microprocessor but only possessing an 8 bit data bus. To do something with 32 bit data required 4 operations. The Gold Card improved on this by using a 68000, which has the same 32 bits internally but also has a 16 bit data bus. Thus only two operations were required for 32 bit data.

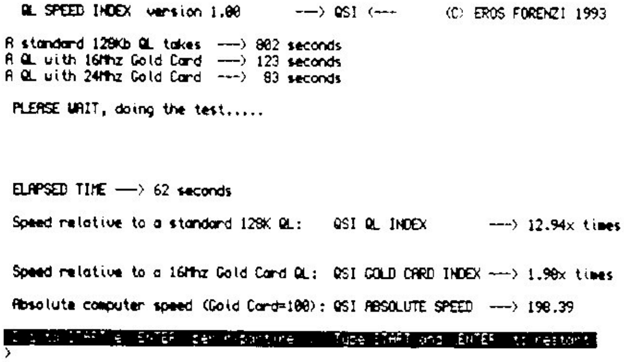
THE SUPER GOLD CARD - (CONT'D)

The Super Gold Card goes one step further by using a 68020 which is a 32 bit cpu both inside and out. Another speedup comes from the system clock speed. The original QL ran at 7.5 MHz, the Gold Card at 16 MHz and the SGC runs at 24 MHz. To add to this, the SGC uses 32 bit RAM compared to the GC's 16 bit RAM and the QL's 8 bit memory. All this combines to make the Gold Card about 4 times faster than the basic QL and the Super Gold Card about 3 times faster than the GC.

Further acceleration can be obtained by taking advantage of the SGC's cache (pronounced 'kash'). A cache is an area of high speed memory reserved for storing frequently used instructions or data. A simple analogy of this can be done with a QL and a microdrive cartridge. The very first time you try to get a directory from a microdrive it will take 5-10 seconds. Once the data on the cartridge is in memory and you ask for another directory, the drive will only spin for a second or so just to check to see if the cartridge is the same. If so, the QL uses the data that it already has in memory which is much faster and therefore the list of files appears much faster. A system using cache memory will check to see if data or an instruction just requested is already in its cache, if so, it will be used from there since it's much faster than regular RAM. There are no complex commands or things to remember about the SGC's cache. Only two commands control it, CACHE_ON and CACHE_OFF. All other control is internal. A way to turn it off has been provided because there are some programs that modify their own code in memory as they run. If the cache was on, the program may execute the old code from the cache rather than the new modified code in memory. The only way to tell is to run your software, if it seems to behave strangely or crashes with the cache on, then turn it off.

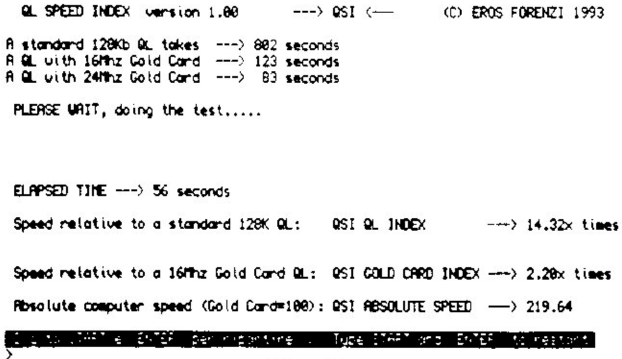
An additional pair of commands control the use of the QL's second screen. Some software uses the second screen and the Minerva ROM makes it easier to use this with your own programs. But a small increase in speed can be realized by disabling any checking for use of the second screen. If you never use it, then turning it off will get the last bit of performance out of the SGC.

Four megabytes of memory may seem like more than you will ever need but the Gold Card's two megabytes seemed the same way and I did, on a few occasions, have a number of programs running at the same time and found little or no free memory left. Now I have some breathing room again. With the extra memory you can really take advantage of QDOS' ability to multitask by having the RAM to load and run more than one of the larger software packages that are becoming available.



```
 QL SPEED INDEX  version 1.00        ---> QSI <---      (C) EROS FORENZI 1993

A standard 128Kb QL takes  ---> 802 seconds
A QL with 16Mhz Gold Card  ---> 123 seconds
A QL with 24Mhz Gold Card  --->  83 seconds

 PLEASE WAIT, doing the test.....



 ELAPSED TIME ---> 62 seconds

 Speed relative to a standard 128K QL:    QSI QL INDEX          ---> 12.94x times

 Speed relative to a 16Mhz Gold Card QL:  QSI GOLD CARD INDEX ---> 1.90x times

 Absolute computer speed (Gold Card=100): QSI ABSOLUTE SPEED  ---> 190.39

[illegible] Type START and ENTER to restore
>
```

(Fig. 1)



```
 QL SPEED INDEX  version 1.00        ---> QSI <---      (C) EROS FORENZI 1993

A standard 128Kb QL takes  ---> 802 seconds
A QL with 16Mhz Gold Card  ---> 123 seconds
A QL with 24Mhz Gold Card  --->  83 seconds

 PLEASE WAIT, doing the test.....



 ELAPSED TIME ---> 56 seconds

 Speed relative to a standard 128K QL:    QSI QL INDEX          ---> 14.32x times

 Speed relative to a 16Mhz Gold Card QL:  QSI GOLD CARD INDEX ---> 2.20x times

 Absolute computer speed (Gold Card=100): QSI ABSOLUTE SPEED  ---> 219.64

[illegible] Type START and ENTER to restore
>
```

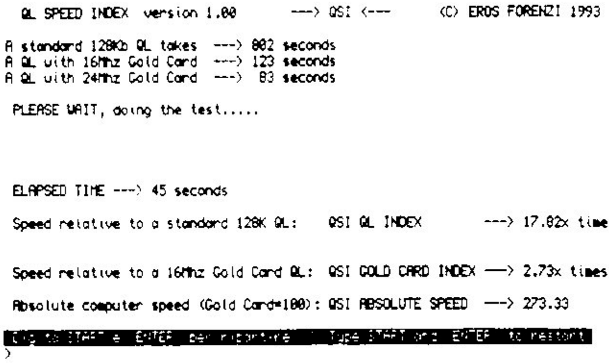
(Fig. 2)

Minerva users have long enjoyed the automatic booting feature of that ROM. Just turn the QL on and it will boot itself up even if you don't press F1 or F2. The SGC now allows this along with the ability to invoke Toolkit 2 at the same time. You have the choice of simulating the pressing of F1 or F2. When used with a Minerva system, the auto boot will be done by Minerva and TK2 will be invoked by the SGC. This can be disabled if you have software that doesn't like TK2 being enabled.

There are many ways to measure the performance of a computer. Some can be quite simple but not give much information, others can be very involved and tell you more than you wanted to know. I used a few different ways to check out my SGC. One of the first things I did was try loading software that I knew ran slowly on a basic QL and then played around with it while running on the SGC. The increase in performance is very noticeable. While not very precise or scientific it does give you immediate feedback especially if you are very familiar with the software and how it runs. At the second Miracle in Newport show, the SGC was demonstrated using Quill! This program is something that probably every QL user is familiar with even if they no longer use it. Remember how scrolling backwards through a long file seemed to take forever? With the SGC Quill takes on a whole new feel.

THE SUPER GOLD CARD - (CONT'D)

To get a little more concrete evidence of the speedup, I turned to a benchmark program written by Eros Forenzi called QL Speed Index (QSI). This should be available from most PD sources, QBox BBS, etc. It is actually an Archive procedure that makes a fixed number of calculations or conversions and times how long it takes and then compares this time to an original 128K QL, and a 16Mhz Gold Card. Since the procedure does not make any file operations, there are no variables such as a slow disk drive. It also gives a 'real world' type of test, in other words, a test made using an actual application (Archive), not some routine that was cooked up to make the computer look good while running one specific test. The instructions state that it should be run on Archive version 2.0 to 2.38 with nothing else loaded into the machine such as TK2, Lightning, other programs, etc. This will give you a good baseline to start from. Then you can load in extensions, toolkits, etc and rerun the Index procedure and see how they affect the performance.

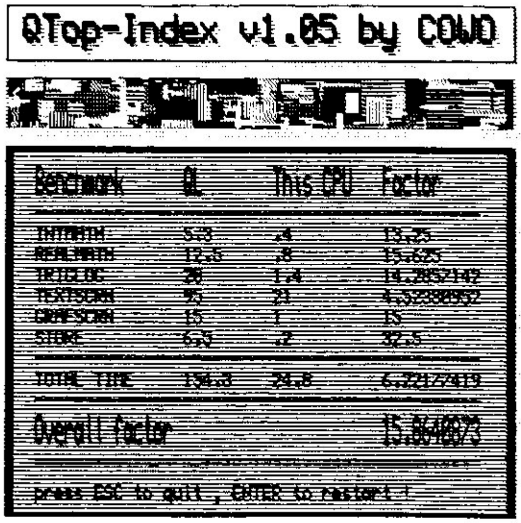


```
 QL SPEED INDEX  version 1.00        ---> QSI <---      (C) EROS FORENZI 1993

A standard 128Kb QL takes  ---> 802 seconds
A QL with 16Mhz Gold Card  ---> 123 seconds
A QL with 24Mhz Gold Card  --->  83 seconds

 PLEASE WAIT, doing the test.....



 ELAPSED TIME ---> 45 seconds

 Speed relative to a standard 128K QL:    QSI QL INDEX          ---> 17.82x times

 Speed relative to a 16Mhz Gold Card QL:  QSI GOLD CARD INDEX ---> 2.73x times

 Absolute computer speed (Gold Card=100): QSI ABSOLUTE SPEED  ---> 273.33
 Digit START e ENTER per ripartire   Type START and ENTER to restart
>
```

(Fig. 3)

You can also see the effects of the SGC's cache and second screen commands on the performance. Figure 1 shows the QSI result from my QL and SGC with the cache off and the second screen enabled. Figure 2 shows the difference that the cache makes. Figure 3 shows the further increase by disabling the second screen.

To get some more performance numbers I turned next to QTOP, a front end package for QDOS computers. QTOP includes its own Index utility that can be used to compare performance differences. This index gets more specific in that it checks speeds in various math routines, text printing to screen, graphics printing to screen, and file storage. Like QSI, it compares the results to an original 128K QL. See figure 4 for the results I got with the cache on, the second screen disabled, and a RAM disk for file storage. A hard disk would be slightly slower and floppies slower still in the store factor.



QTop-Index v1.05 by COWD

Benchmark	QL	This CPU	Factor
INTMATH	5.3	.4	13.25
REALMATH	12.5	.8	15.625
TRIGLOG	20	1.4	14.2857142
TEXTSCRN	95	21	4.52380952
GRAFSCRN	15	1	15
STORE	6.5	.2	32.5
TOTAL TIME	154.3	24.8	6.22177419
Overall factor			15.8640873

press ESC to quit , ENTER to restart !

The Super Gold Card builds on the original Gold Card's features by adding more memory and speed and new features like the parallel port. It is very well designed and built and has been completely reliable. Adding a Super Gold Card to a QL and using the combination makes it hard to believe that the QL is a ten year old computer! The Super Gold Card makes the QL feel like a whole new machine.

NEWS From DIGITAL PRECISION

Freedy Vachha of Digital Precision would like the readers of IQLR to know that as of the 10th of October 1994 DP has NO orders, refunds due or correspondence outstanding. He would also like to thank the loyal DP customers who put up with delays caused by developments on the personal front.

Due to an increasingly heavy work load (the job that pays the bills) DP will not be able to provide on-line phone service until further notice, orders and queries by letter will be handled expeditiously.

QXL In COMMAND

Bedford, Massachesetts & Pylesville, Maryland, USA
Al Boehm *Tom Robbins*

Note to readers: The 'I' used in this article is sometimes Al and sometimes Tom. This series is assembled via E-Mail as a joint production.

QXL WINCHESTER HARD DISK - The QXL uses the PC's hard disk by formating a segment of it for its own use. The segment can be 1 up to 66 Mbytes in size. The QXL manual warns not to go higher then 63 but I (Al) have successfully formated 66 Mbytes on several computers without any trouble - so far. 67 Mbytes won't work. The QXL formated section shows up on the PC directory as QXL.WIN.

The QXL command FORMAT "WIN1_1" will format a 1 Mbyte section of the PC's hard disk for use by the QXL. It will take about half a minute and return with 512/512 sectors which is apparently a harmless bug since the command DIR WIN1_ gives the corect 2032/2048 sectors. A QL sector is 512 bytes so that 2048 * 512 = 1,048,576 or in computer language, 1 Mbyte. The first number, 2032, is the number of sectors available which will decrease when files are written to WIN1. The 8 sectors difference between 2032 and 2048 are used for file header data.

All the SBasic and TK2 file commands such as DIR, OPEN, DELETE, WSTAT, work the same as on microdrives or floppies except WIN1_ is used instead of MDV1_ or FLP1_. Reading and writing files is, of course, much much faster. The PC uses different header files so trying to do anything useful such as reading or copying QXL.WIN from the PC has not been sucessful. DEL QXL.WIN to delete does work; so be careful!

With the current QXL operating system, only up to 66 Mbytes can be formated for WIN1. However, if you have two harddisks, FORMAT "WIN2_66" will set up a WIN2 on the second harddisk.

Further, it is possible with a PC to partition a single hard disk using the PC FDISK command. Make sure you back up anything on the hard disk before using this command; its possible to lose it. With the additional partition(s) you can have a WIN2, WIN3, etc.

It is also possible to use the 'partition' created by disk compression software such as MSDOS Doublespace and Drivespace as an additional QXL hard disk. If the PC hard disk does not have enough room for the amount asked for by the QXL FORMAT the format will occur with whatever room there is. Nice touch Miracle!

QXL RAM DISK ON THE PC - If the PC has a 1 Mbyte ramdisk initiated, then the QXL can use it by the command FORMAT WIN2_1 (WIN3 or WIN4 etc. when there is more than one hard disk or partition. This format takes less then a second. For additional hard disks or PC ram disks, the PC drive c: is win1, d: is win2, etc. Reading and writing to this ram disk is quite fast. Of course anything written to win2 is gone when the computer is turned off.

A ram disk large enough for a 1Mbyte QXL WIN device is initiated on a PC by including the command, DEVICE=C:\DOS\RAMDRIVE.SYS 1030 /E in the CONFIG.SYS file after the HIMEM.SYS command. There are a lot of options so read the PC manual or help file. The number 1030 means to set aside 1030 Kbytes for the ram disk. Why this isn't 1048 beats me but 1030 is large enough to do the job. Up to 4 M byte ram disks can be initiated on my PC using DOS 6.1.

The PC has trouble reading the QXL WIN formated ram disk. But I was able to read (with some gibberish) a small (10K) ram disk using the PC EDIT program. Furthemore, the PC should be able to PEEK and POKE into the ram disk storage area thus allowing fast communication between the PC and the QXL.

USE DIRECTORIES - With dozens even hundreds of programs on a hard disk, it highly recomended that you use directories to store things in a logical manner or you will have a tough time finding them (this is experience speaking). Use the command MAKE_DIR "WIN1_games_" to make a directory called games. When you type DIR WIN1_, games shows up as games -> indicating it is a directory.

QXL In COMMAND - (CONT'D)

DO BOOTS - Even with directories, I forget where things are and have a proliferation of programs called BOOT. I solved this problem by puting all my boot programs in a seperate directory called BOOTS. I give each one a name that is short and easy to remember. For example, I use "text" for the Text87plus4 boot. My QXL boot contains TK2_ext and PROG_USE WIN1_BOOTS_ to start Toolkit2 and make BOOTS the default device for executing programs. To start Text87plus4, I simply type DO text.

I can now start every thing with a DO. I don't have to remember whether a program is started with EX or EW or LRUN or LRESPR or whatever. That is all set up in my boot programs in BOOTS. I just type DO it and "it" is done! Some programs can be started with different options (for example window size). I give each option that I use a diferent DO boot with a unique name. I plan to send a fuller description of this DO BOOTS idea into QUANTA; look for it.

5 1/4 inch 720 K Drives and the QXL - IBM PC's and compatables do not officially support this format. That is, if you have an IBM 1.2 MByte 5 1/4 inch drive it will not read or write the 720 K bytes "standard" QL 5 1.4 inch floppies. What then if some or all of your QL files are in this format?

There are two options:

1. The network ports on the QXL are fully operational. Assuming you have a QL with Toolkit 2, you can network your QL to the QXL and use the net to access the 5 1/4 inch drives.

2. If your PC has only one disk drive, you CAN install a QL compatable 720 K 5 1/4" drive. The standard PC disk controller only supports 2 floppy drives. The PC does not know the difference between a 3 1/2" and a 5 1/4" 720 K drive. If you include the line "drivparm= /d:1 /f:2" in the pc's config.sys file, this installs a 3 1/2" 720 K drive as drive B:, but MSDOS doesn't care whether it is a 5 1/4" or 3 1/2" drive. This drive can be used for either the QXL as FLP2_ or the PC as drive B:.

I have tried both of these methods and both work. The advantage of method 1 is that if you set up both the QL and the QXL as servers with the FSERVE command, you can now use the QXL hard disk from your QL as n1_win1_.

Note that the QL 3 1/2 inch double density floppies work just fine in the QXL. Moreover, if a 1.4 M Bytes (i.e. High Density) 3 1/2 inch floppy is formated by the QXL it will format to 1.4 M Bytes and read and write from the QXL but of course won't work in the standard 720 K bytes QL drive.

SCREEN DISPLAY - The QL mode uses only part of the screen. On a 15" monitor when the QXL is in QL mode the display is about the same size as on my QL's 14" monitor. The size of letters in the EGA mode is the same but more can be written across if the software windows can be adjusted.

As mentioned in an earlier article, programs that use PEEK or POKE to write to the screen memory directly do not run correctly in EGA mode, but run quite well in QL mode. Lightning from Digital Precision is an example, as is TEXT87 plus4.

CRIMES AGAINST NATURE AND UNSPEAKABLE ACTS - It is possible to run Conqueror on the QXL and thus have a PC supporting a QXL emulating a PC. Why anyone would want to do this, I don't know - but I just had to try it.

BENCHMARKS - The timings obtained on the QXL running on a 66 MHZ 486DX PC were essentiallly the same as those obtained on a 10 MHZ XT as reported in our previous article. Hard drive access was considerably better, as the XT had a very slow hard drive. Additionally, screen handling appears better on the faster PC, which has a local bus video system.

The QXL runs quite well under both Windows and DOS. In fact, I have even had it running in a 'DOS box' on the windows screen, albeit slowly. I will have some timings on the QXL under DOS versus the QXL under Windows for the next article in the series.

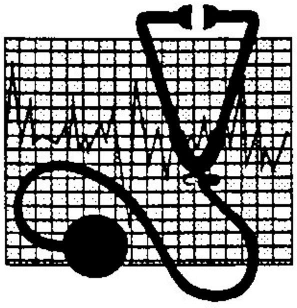
QXL In COMMAND - (CONT'D)

Lightning from Digital Precision loads and runs under QL mode, but does not operate correctly under EGA mode. What effect Lightning has on screen timings will also be reported in the next article in this series.

There IS a way to use a 1.2M floppy drive on a PC as a 720K drive for the QXL that will read and write floppy disks that can be read on a QL with 5 1/4" 720K drive. Further details in next issue.

QL SPIN DOCTOR

Troy, Michigan, USA - Don Walterman



This issue I'd like to discuss Sony ED drives. First though, a few updates to last issues firmware lists for disk interfaces. Update the Miracle Super Gold Card firmware to 2.49. Apparently there is a version 1.17 Cumana rom in existence. If anyone has access to one please let me know. Also add QUBIDE to the list of Hard Disk interfaces. QUBIDE is currently shipping with version 1.28 firmware (see review in this issue).

ED floppies have been a challenge for Gold Card and Super Gold Card owners. The QL needs floppies that can be configured to meet the QL requirements. Fortunately we have managed to determine the Teac ED jumper configurations and have published them in previous issues. Sony ED drives are frequently available at low prices. I finally bought a couple and found out they are really difficult drives to use. There appear to be a whole range of Sony ED drives. Only a few will work with the QL. These drives do not have jumpers so pay close attention to the exact model numbers. The Sony ED drives usually are labelled '2.88' on the disk eject button. The Sony drives have numbers all over the housing. The actual model number is attached to the cover over the motor on the bottom of the drive. It should be of the format SONY Model MP-F40W-XX. The XX portion of the number is where the fun begins. I have found out about the following drives:

XX=	13	Real Problems. Can work in HD or ED mode only.
XX=	15	Should work according to Sony but not tested.
XX=	23	Does work and has been tested on QLs.
XX=	27	Should be the same as 23 but not tested.

As you can see the only safe drive to buy is the MP-F40W-23. I got the XX=13 version at a very cheap price. After many calls to Sony technical support,.I got a little information. By the way, Sony technical support leaves you on hold forever, drops you into their voice mail and rarely calls back. It took me months of calling to finally talk to a human.

The Sony technical service person told me the XX=13 model was custom made for a big computer company (either NCR or NEC) to their specifications. Sony was under a non-disclosure deal so they could not tell me much about the drive. The only information they would give me was that the drive did not have an 'auto-TTL' function whatever that may be. Apparently the XX=13 drive does not look at the holes punched in the disk to mark the density. It tried to format all my disks as ED disks. It also tried to read everything as an ED disk. After some major hair-pulling, I found that by grounding pin 34 the drive would format everything inserted as an HD disk. These formatted disks were compatible with drives on my other systems.

So, the bottom line is to make the MP-F40W-13 marginally useful attach a small switch between pin 33 (ground) and pin 34. Opening the connection will tell the drive to read and write ED disks always. Closing the connection will tell the drive to read and write all disks as HD disks. I haven't figured out any way to read or write standard DD disks with this drive. The switches that sense HD or ED holes are physically present on the drive so my next step will be to wire one of them directly to pin 34. If you only use HD and ED disks and find this drive you may be able to use it. Otherwise, you have been warned.

I'd like to hear anyone else's experiences with other vendors ED drives. I know a number of other vendors sell them including Mitsumi, Alpine and Epson. If you have any information or corrections you would like to share, please contact me in care of IQLR or leave me a message on your nearest QBox BBS.

Professional & Graphical Softwa

NEW SOFTWARE FROM BELGIUM

Amelia Iowan Old Style BREMEN BOLD Kuenstler 480 ∞ΔΓαβ Goudy Handtooled
Caslon Ope[illegible] ian Bodoni
Revival 565 SHOTGUN
Shelley Alleg[illegible] Staccato 555
Venetian 30[illegible] [illegible]agull Bold
Calligraphi[illegible] [illegible]el Gothic
Carmina Bold Italic [illegible] INFORMAL 011 Lydian UMBRA [illegible] 591 Allegro
Old Dreadful No.7 Goudy Old Style Freeform 721 Dom Casual Amerigo Carmina Italic

PROforma fontpack

A collection of 100 high quality Bitstream® fonts

All fonts have high quality outlines, a full charset, hinting and kerning !

PFdata

Very interesting program for all DATAdesign owners. This program can create hardcopy of your DATAdesign files using PROforma. This means that you can use a large selection of fonts (as the ones included in LINEdesign), in any requested size. Also LINEdesign pictures can be included to add logo's, boxes etc. Several records can be printed on each page, several options for justification etc.

PFlist

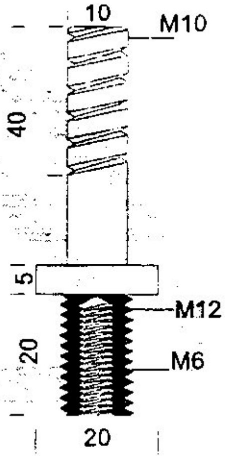
Very easy to use program to create listings on any printer (especially inkjet and laser printers). Can include a footer with filename and filedate. Always allows perforation of your pages. The font and fontsize can be chosen (PFlist uses PROforma). PFlist can create your listing in two columns, and landscape (or both).

pfb2pff

This program gives you access to a large variaty of fonts for PROforma, LINEdesign, PFdata, PFlist, ...
It allows you to convert Adobe Type 1 _pfb fonts to PROforma.

AND OUR ALL TIME FAVORITES

LINEdesign



Our definitive favorite. LINEdesign is a very powerful vector drawing program. It is very userfriendly and can be used by everybody (as long as you have access to a QL or compatible computer). It includes many fonts and clipart drawings. LINEdesign can create pictures which can be reproduced on any printer and at any size, with the same result (resolution permitting).
LINEdesign is very flexible. It can be used for artistic drawing, technical drawing, creating adverts (like this one), leaflets, fliers... LINEdesign can even do some desktop publishing.
If you want high quality output, then LINEdesign is the program for you !
LINEdesign allows you to create pages with text, lines, curves, rectangles (with rounded corners), pies, bitmaps,... These can all be scaled, rotated, slanted, moved without degeneration. LINEdesign is delivered with an extensive manual including a printout of all fonts and clipart

PROforma

The graphics library which we developped for LINEdesign. This library can be used by C programmers, and gives you access to all drawing commands and printer drivers which are used by LINEdesign, PFdata and PFlist, and even more !!

DATAdesign

A very userfriendly database which is powerful and fast. The userfriendlyness of this program is quite amazing. Can be used for all you database requirements.

DATAdesign API

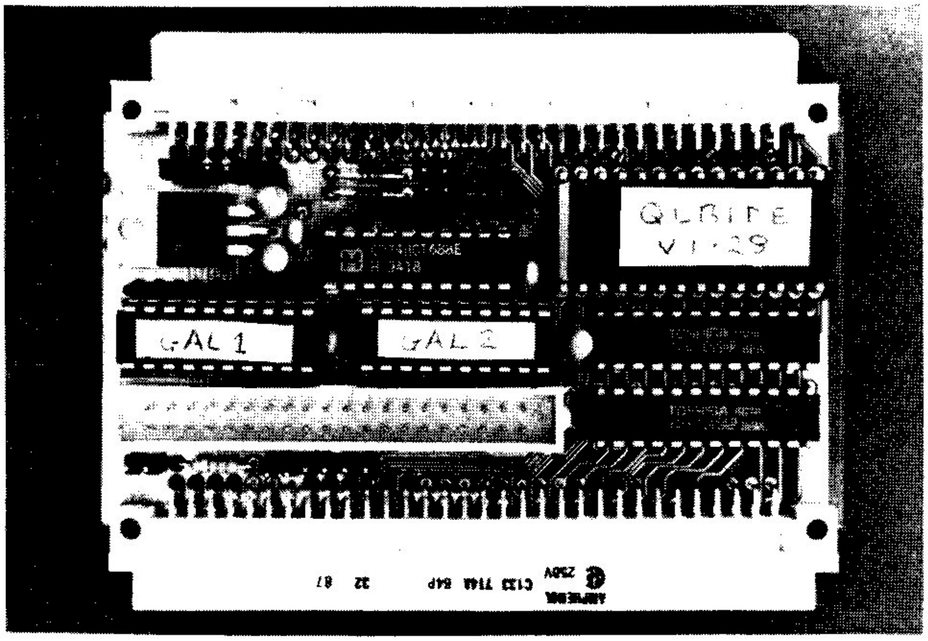
Unleash the real power of DATAdesign. The API gives programmers direct access to the DATAdesign database management system. It offers you a record at a time data manipulation language which can be used from SuperBasic, C and Assembler.
When you access the DATAdesign engine directly, DATAdesign suddenly turns into a fully relational database with an extra, as all records can always be referenced in a unique way.
DATAdesign and the API allow you to use variable length fields. Fields can be added or deleted witthout problems. Files can be accessed by several jobs at the same time etc.

for more information, just write, phone or fax !!
Haachtstraat 92, 3020 Veltem, Belgium, tel/fax (016) 48 89 52

PROforma fontpack BEF 4000 (3520), PFdata BEF 1000 (930), PFlist BEF 1000 (930), pfb2pff BEF 3000 (2700), LINEdesign BEF 5000 (4350), PROforma BEF 5000 (4350), DATAdesign BEF 3000 (2700), DATAdesign API BEF 1000 (930). Prices in brackets for customers outside European Community.
Payment terms : postage is included, send a Eurocheque in BEF, or your VISA/ EuroCard/ MasterCard number and expiry date. For updates and upgrades, inquire !

QUBIDE

QL AT/IDE Interface



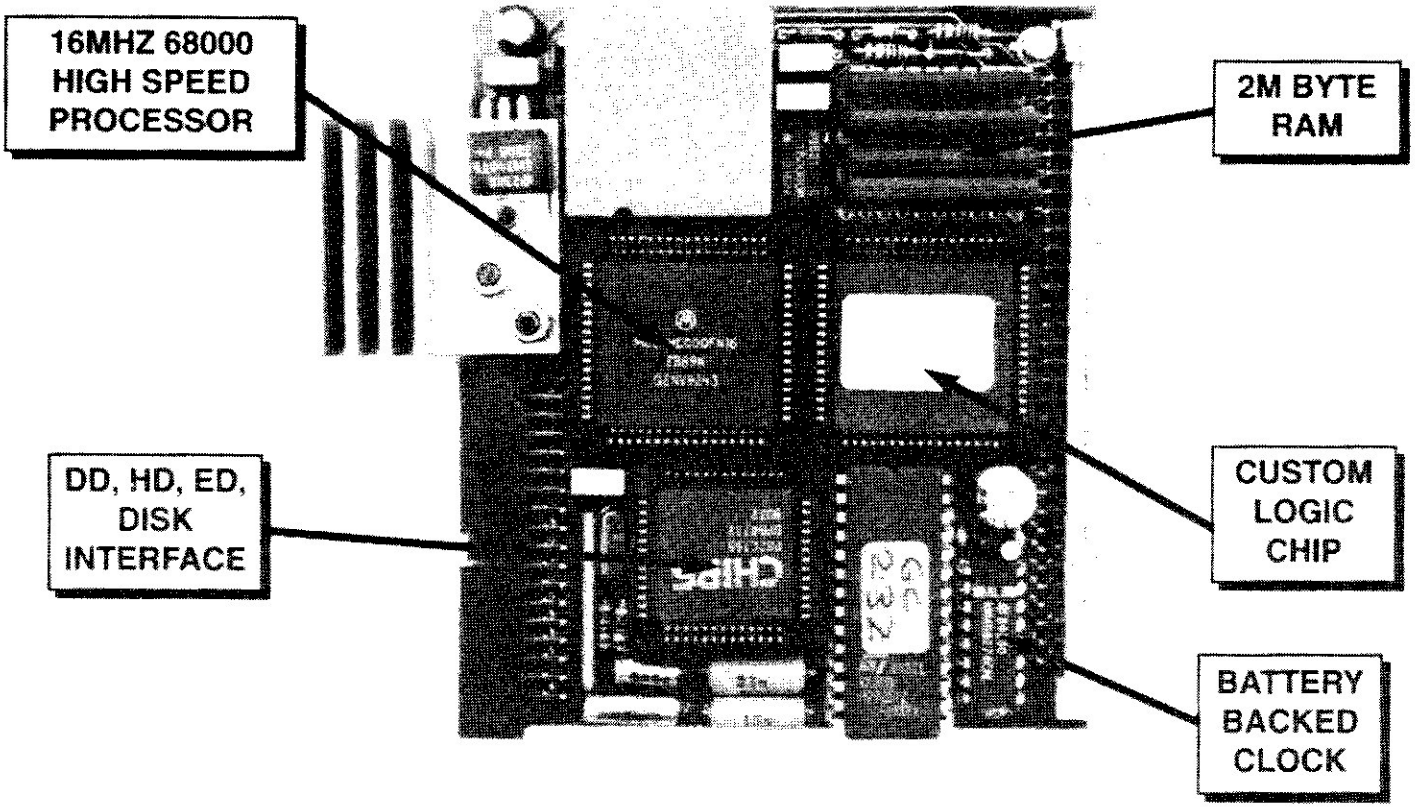
QUBIDE brings the Sinclair QL into the 90's, It revolutionises the meaning of Mass Storage Media on the QL. With QUBIDE you are now able to connect modern AT/IDE Hard Drives to your Sinclair QL. A massive amount of storage space can now be made available for your programs and files. QUBIDE is fully compatible with Super Gold Card, Gold Card, Trump Card and most memory expansion systems for the QL, also Minerva and Hermes compatible.

To Order
Please make cheques payable to
QUBBESoft P/D in Pounds Sterling.

QUBBESoft P/D
38, Brunwin Road, Rayne,
Braintree, Essex. CM7 5BU. UK
Tel: 0376 347852 Fax: 0376 331267

QUBIDE Price
£65.00p
fully inclusive of P&P (UK)
+5% (Europe), +10% (Rest of World)
Also includes 1 year Warranty

MIRACLE SYSTEMS



QL GOLD CARD

Recycled Gold Card £150 inc. (£125 outside EC)

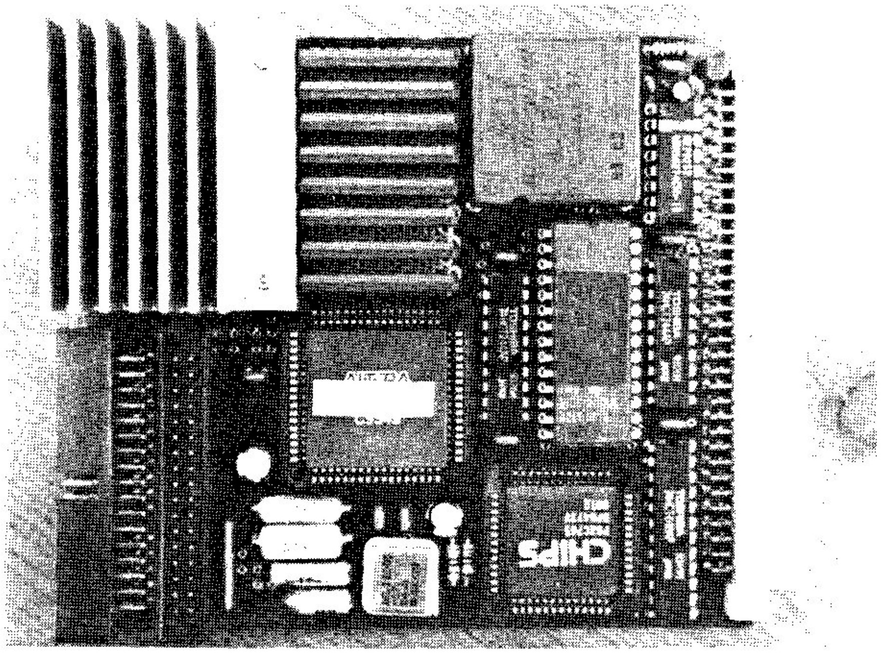
This is the expansion that has been revolutionising the QL. It is very easy to fit , it simply plugs into the expansion port at the left hand of the QL, and once fitted it will instantly increase the execution speed of the QL by about 4 times due to the presence of a 16MHz 68000 on board. There is 2M of fast 16 bit RAM of which QDOS sees a contiguous 1920K. The remainder is used for shadowing the QL's ROM and display memory and for the GOLD CARD's own code.

There is a disk interface which can access 3 mechanisms (4 with the DISK ADAPTER) of three different densities, DD (double density, 720K), HD (high density, 1.44M) and ED (extra high density, 3.2M) in any mix. The disk interface connector is the same type that was fitted to the Trump Card so most QL compatible disk drives can be used.

> Please note: that DD drives still give a capacity of 720K per diskette.
> Our DUAL ED DISK DRIVE allows the GOLD CARD to access DD, HD and ED diskettes.

Another feature is the battery backed clock. When the QL is switched on the contents of the clock are copied into the QL's clock so that the time and date are correct. The firmware in the ROM gives the GOLD CARD all the functionality of the Trump Card like TOOLKIT II and there is a sub-directory system for floppy and RAM disks.

Physically the GOLD CARD is about half the size of the TRUMP CARD and so fits almost all within the QL. Its current consumption is well under allowable maximum so no special power supply is required. The GOLD CARD comes with a 14 day money back guarantee and a 1 year warranty.



MIRACLE SYSTEMS

See below for GOLD CARD upgrade deal !

Call for other upgrade paths.

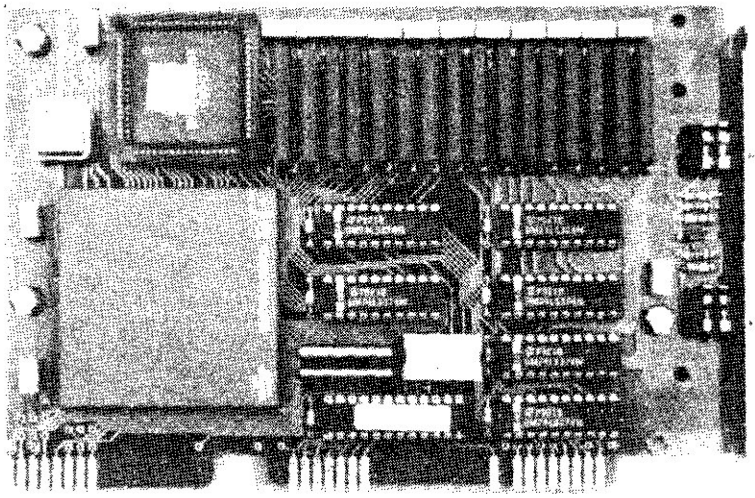
QXL

Now With

SBASIC

SuperBasic Compatible Interpreter

Please Note:
Our NEW address and
Telephone/Fax number
listed below.



- *68EC040*
- *4M or 8M of RAM*
- *Multitasking SBASIC*
- *QL Network ports*
- *Toolkit II*
- *QDOS or MSDOS floppies*
- *Uses PC's keyboard, floppy & hard disks, parallel/serial ports and mouse.*

This is the card that plugs into a standard 8 or 16 bit ISA slot on a PC and allows the PC to run QL programs - FAST. A new QDOS compatible operating system from Tony Tebby called SMSQ, which is supplied on a disk, includes Toolkit II and gives you the familiar QL environment. SMSQ includes SBASIC a multitasking SuperBasic compatible interpreter.

Installation is simple; plug the QXL into a spare slot and copy 2 files from the supplied disk onto the hard drive and you're ready to go. From the DOS prompt type QXL and the PC will transform itself into a QL before your very eyes. If at any stage you wish to return to DOS just press CTRL-ScrollLock. You can later resume the QL session by typing QXL/ which takes you back to where you left off.

For POINTER ENVIRONMENT programs SMSQ can be configured to handle 3 screen resolutions in addition to the standard 512x256 QL screen. Your PC must have EGA or VGA graphics. EGA allows 640x350 whereas VGA also allows 640x480. Most SVGA cards will allow SMSQ to use 800x600 as well.

A QXL fitted with 4M bytes of RAM costs £380 including VAT (£330 outside the EU) and the 8M byte version is £495 (£430 outside the EU). Prices include postage, our 2 year warranty and a 14 day money back guarantee. Software updates are supplied free of charge and sent out automatically.

Alternatively send us your GOLD CARD plus £230 (£205 outside the EU) for a 4M byte QXL or send us your GOLD CARD and £345 (£305 outside the EU) for an 8M byte QXL.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

TELEPHONE/FAX: (01454) 883 602

MIRACLE SYSTEMS LTD - 20 Mow Barton, Yate, Bristol, BS17 5NF, UK